

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

«На правах рукопису»
УДК 004.93

«До захисту допущено»

Завідувач кафедри
_____ І.Р. Пархомей
(підпис)

“ ____ ” _____ 2019 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 126 «Інформаційні системи та технології»

на тему: Підсистема розпізнавання об'єктів відеопотоку інтелектуального
робота

Виконав (-ла): студент (-ка) другого курсу, групи ІК-81МП
(шифр групи)

Южда Юлія Віталіївна

(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник ст. викладач Анікін В.К.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант НК

(назва розділу)

доцент Пасько В.П.

(науковий ступінь, вчене звання, прізвище, ініціали)

(підпис)

Рецензент _____

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2019 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Рівень вищої освіти – другий (магістерський)

Спеціальність 126 «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ І.Р. Пархомей

(підпис)

« ____ » _____ 2019 р.

**ЗАВДАННЯ
на магістерську дисертацію студенту**

Южді Юлії Віталіївні

(прізвище, ім'я, по батькові)

1. Тема дисертації Підсистема розпізнавання об'єктів відеопотоку інтелектуального робота

науковий керівник дисертації _____ ст. викладач Анікін В.К. _____ ,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від « 28 » жовтня 2019 р. № 3770-с

2. Термін подання студентом дисертації _____ 9 грудня 2019р.

3. Об'єкт дослідження комп'ютерний зір для розпізнавання об'єктів за даними відеопотоку

4. Предмет дослідження відеоспостереження за об'єктами за умови сильно зашумлення, зміни інтенсивності освітлення

5. Перелік завдань, які потрібно розробити аналіз точності та ефективності існуючих методів та алгоритмів розпізнавання об'єктів, обрати метод відповідно до вимог підсистеми та за рахунок комбінованого використання фільтрації шуму, морфологічних операцій, гама-корекції та багатопоточності підвищити точність розпізнавання, розробити програмне забезпечення підсистеми розпізнавання та провести її тестування, використовуючи різні

варіанти
даних

неякісних

вхідних

6. Орієнтовний перелік ілюстративного матеріалу
6 плакатів

7. Орієнтовний перелік публікацій XV Міжнародна науково-практична конференція «Нові досягнення в європейській науці» (Болгарія, 15 - 22 червня 2019 р.), XV Міжнародна науково-практична конференція «Наукові горизонти-2019» (Англія, 30 вересня – 7 жовтня, 2019)

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
НК	Пасько В. П., доцент		
Перевірка на співпадіння	Лісовиченко О. І., доцент		

9. Дата видачі завдання 26 вересня 2018 року

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Формування проблематики	02.09.2019 – 08.09.2019	
2	Аналіз проблематики	09.09.2019 – 15.09.2019	
3	Постановка задачі	16.09.2019 – 22.09.2019	
4	Аналіз існуючих методів та алгоритмів	23.09.2019 – 29.09.2019	
5	Вибір засобів для створення підсистеми	30.09.2019 – 06.10.2019	
6	Розробка підсистеми	07.10.2019 – 13.10.2019	
7	Тестування та покращення підсистеми	14.10.2019 – 20.10.2019	
8	Практичне застосування ПЗ	21.10.2019 - 27.10.2019	

Студент

(підпис)

Ю. В. Южда
(ініціали, прізвище)

Науковий керівник дисертації

(підпис)

В. К. Анікін
(ініціали, прізвище)

АНОТАЦІЯ

Актуальність теми. В останній час широке розповсюдження отримує відеоаналітика - технологія, яка використовує комп'ютерний зір для автоматизованого збору інформації у послідовності кадрів, які отримуються з відеокамер в реальному часі чи з відеозаписів. Дана технологія може бути застосована в відеоспостереженні, системах безпеки, транспорті.

Об'єктом є комп'ютерний зір для розпізнавання об'єктів за даними відеопотоку.

Предметом дослідження є алгоритми і методи виявлення об'єктів в кадрі та безпосереднє розпізнавання знайдених об'єктів.

Мета роботи: підвищення якості процесу розпізнавання об'єктів відеопотоку шляхом використання модифікованого методу ідентифікації.

Наукова новизна полягає в наступному: удосконалено процес пошуку об'єктів відеопотоку, за рахунок комбінованого використання фільтрації шуму, гама-корекції, морфологічних операцій та використання багатопоточності для розпізнавання, який має здійснювати розпізнавання об'єктів для різних варіантів неякісних вхідних даних в режимі реального часу.

Практична цінність розроблена система може використовуватись при вирішеннях різних завдань відеоаналітики.

Апробація роботи. Основні положення і результати роботи були представлені на XV Міжнародній науково-практичній конференції «Нові досягнення в європейській науці» (Болгарія, 15 - 22 червня 2019 р.) та на XV Міжнародній науково-практичній конференції «Наукові горизонти-2019» (Англія, 30 вересня – 7 жовтня, 2019).

Структура та обсяг роботи. Магістерська дисертація складається з вступу, чотирьох розділів та висновків.

У вступі подано загальну характеристику роботи, зроблено оцінку сучасного стану проблеми, обґрунтовано актуальність напрямку

досліджень, сформульовано мету і задачі досліджень.

У першому розділі розглянуто існуючі методи виявлення та розпізнавання об'єктів, проаналізовано і поставлено задачу розпізнавання об'єктів, описано функціональні вимоги до розроблюваної системи.

У другому розділі описуються основні методи відстежування об'єктів у відеопотоці, детально описана проблема перекриття об'єктів на сцені.

У третьому розділі зроблено вибір інструментарію розробки, детально описується програмна реалізація системи, проводиться її тестування.

У четвертому розділі було здійснено маркетинговий аналіз стартап проекту.

У висновках представлені результати проведеної роботи.

Робота представлена на 94 аркуші, містить 25 рисунків, містить 20 посилань на список з використаних літературних джерел.

Ключові слова: розпізнавання об'єктів, відеопоток, відстежування об'єктів, оклюзія, гама-корекція, морфологічні операції.

ABSTRACT

Actuality of theme. Recently, video analytics has become widespread - a technology that uses computer vision to automate the collection of information in a sequence of frames obtained from live video cameras or from video recordings. This technology can be applied in CCTV, security systems, transport.

The object is a computer vision for recognizing objects by video stream data.

The subject of the study are algorithms and methods for detecting objects in the frame and direct recognition of found objects.

Purpose: Improve the quality of the video stream object recognition process by using a modified identification method.

The scientific novelty is:

The process of searching for video stream objects has been improved by combining noise filtering, gamma correction, morphological operations and the use of multithreading for recognition, which should recognize objects for different variants of low-quality real-time input.

The practical value of the developed system can be used in solving various problems of video analytics.

Testing the work. The main provisions and results of the work were presented at the XV International Scientific and Practical Conference "New Advances in European Science" (Bulgaria, June 15 - 22, 2019) and at the XV International Scientific Conference "Scientific Horizons-2019" (England, 30 September - October 7, 2019).

Structure and scope of work. The master's thesis consists of an introduction, four sections and conclusions.

The introduction describes the general characteristics of the work, made an assessment of the current state of the problem, substantiated the relevance of the research direction, formulated the purpose and objectives of the research.

The first section examines the existing methods of object detection and

recognition, analyzes and sets the task of object recognition, describes the functional requirements for the developed system.

The second section describes the basic methods of tracking objects in the video stream, describes in detail the problem of overlapping objects on stage.

In the third section the choice of development tools is made, the software implementation of the system is described in detail, its testing is carried out.

In the fourth section, a marketing analysis of the startup project was carried out.

The results of the work carried out are presented in the conclusions.

The work is presented on 94 sheets, contains 25 drawings, contains 20 links to the list of used literature sources.

Keywords: object recognition, video stream, object tracking, occlusion, gamma correction, morphological operations.

Пояснювальна записка
до магістерської дисертації

на тему: *«Підсистема розпізнавання об'єктів відеопотоку інтелектуального
робота»*

Київ – 2019 року

ЗМІСТ

ВСТУП	11
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ.....	14
1.1 Особливості і проблематика	14
1.1 Способи зображення об'єктів	15
1.3 Виділення характерних рис для відстежування.....	17
1.4 Оптичний потік і його алгоритми.....	17
1.5 Методи виявлення об'єктів	24
Висновки за розділом.....	31
2 ВІДСТЕЖЕННЯ ОБ'ЄКТІВ У ВІДЕОПОТОЦІ	32
2.1 Відстеження об'єктів у відеопотоці.....	32
2.2 Проблема перекриття при відстежуванні об'єктів.....	41
Висновки за розділом.....	48
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПІДСИСТЕМИ.....	50
3.1 Інструментарій розробки.....	50
3.2 Аналіз бібліотеки OpenCV	54
3.3 Морфологічні зміни зображення	56
3.4 Модифікація алгоритму розпізнавання об'єктів	59
3.5 Підвищення якості відеотрекінгу за рахунок використання багатопоточності	63
3.6 Тестування розробленої підсистеми	66
Висновки за розділом.....	72
4. РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ.....	74
4.1 Опис ідеї проекту	74

4.2 Технологічний аудит ідеї проекту	76
4.3 Аналіз ринкових можливостей запуску стартап-проекту	77
4.4 Аналіз ринкової стратегії проекту	82
4.5 Розроблення маркетингової програми стартап-проекту	85
ВИСНОВКИ.....	89
ПЕРЕЛІК ПОСИЛАНЬ	90
ДОДАТКИ.....	92
ДОДАТОК А.....	93
ДОДАТОК Б	94

ВСТУП

Визначення об'єктів, що рухаються є важливою задачею у сфері комп'ютерного зору для різних роботизованих пристроїв. Комп'ютерний зір включає в себе отримання цифрового зображення, його обробку, аналіз і розуміння зображень, використовуючи статистичні методи і моделі побудовані за допомогою фізики, геометрії, статистики і теорії статистичного навчання. Завдання розробки алгоритмів систем відеоспостереження в даний час є надзвичайно актуальною і зачіпає багато сторін життя людини. З появою дешевих цифрових відеореєстраторів стало можливим обробляти вимірювальні дані за допомогою комп'ютера. Прикладами таких завдань є охорона периметра і внутрішньої території об'єкта, виявлення і розпізнавання осіб, розпізнавання руху. Системи відеоспостереження знаходять широке застосування в задачах боротьби з тероризмом і моніторингу дорожнього руху. Великий інтерес в області алгоритмів супроводження руху зумовило поширення потужних комп'ютерів, доступність високоякісних камер за невеликою ціною та збільшення потреб для автоматизованого аналізу відео. Основні ключові кроки відеоаналітики: розпізнавання об'єкта, що рухається, відстеження об'єкта кадр за кадром, і аналіз об'єктів для визначення їх поведінки. Таким чином, використання алгоритму стеження за об'єктом є доречним в задачах:

- розпізнавання на основі руху;
- автоматизоване спостереження, що виявляє підозрілу активність;
- взаємодія людина-комп'ютер, наприклад, слідування за поглядом для вводу даних, розпізнавання жестів;
- навігація машин, що пов'язана з планування маршруту та униканням перешкод.

Наукова новизна роботи полягає в наступному: удосконалено алгоритм пошуку об'єктів відеопотоку, за рахунок комбінованого

використання фільтрації шуму, гама-корекції, морфологічних операцій та використання багатопоточності для розпізнавання, який має здійснювати розпізнавання об'єктів для різних варіантів неякісних вхідних даних в режимі реального часу.

Відстежування об'єктів є складною задачею через такі фактори:

- нестача інформації через проекцію трьохвимірного світу на зображення;
- шум в зображеннях;
- складні рухи об'єктів;
- часткова або повна оклюзія об'єктів;
- зміна освітлення;
- потреба у визначенні об'єктів у реальному часі.

Динамічні методи обробки зображень отримали в даний час великий розвиток, що пов'язано зі значним підвищенням потужності обчислювальної техніки, що дозволяє в даний час ефективно обробляти великі потоки даних в реальному часі.

У зв'язку з цим представляється актуальною розробка автоматизованих методів динамічної сегментації для розпізнавання рухомих об'єктів.

Мета роботи – підвищення якості процесу розпізнавання об'єктів відеопотоку шляхом використання фільтрації шуму, морфологічних операцій та винесенням зчитування вхідних даних у окремий потік. Для досягнення поставленої мети необхідно:

1. провести аналіз (ефективності та точності) існуючих методів виділення і розпізнавання об'єктів;
2. обрати метод розпізнавання відповідно до вимог та за рахунок комбінованого використання фільтрації шуму, гама-корекції, морфологічних операцій та використання багатопоточності підвищити точність розпізнавання;
3. розробити програмне забезпечення розпізнавання об'єкта в

відеопотоці на мові програмування Python;

4. провести випробування розробленого програмного забезпечення для різних варіантів неякісних вхідних даних.

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

1.1 Особливості і проблематика

Основні проблеми, які виникають при створенні та роботі відстежувача, пов'язані з виглядом об'єкта та схожими іншими об'єктами на сцені. Вигляд інших об'єктів може накладатися на задній фон [1]. В таких випадках, важко знайти на зображенні об'єкт, що як ми очікуємо, буде рухатися. На рис. 1.1 зображено приклад помилкового визначення об'єкту.



Рисунок 1.1 – Помилкове визначення об'єктів

Також важко визначити об'єкт, вигляд якого змінився в площині об'єктива через такі фактори:

1. зміна позиції, об'єкт що рухається змінює свій вигляд на площині зображення, наприклад якщо крутиться;
2. зміна освітлення, напрямок, інтенсивність, колір впливає на вигляд об'єкта. Так, зміна світла в глобальному плані викликає проблеми у

сусідніх сценах. Це впливає на те, як ми бачимо зображення через лінзу камери;

3. процес отримання зображення пов'язаний з певною долею шуму, який залежить від якості матриці камери;
4. спостерігати за ціллю, коли вона частково або повністю перекрита іншим об'єктом на зображенні.

Перекриття виникають коли:

- ціль рухається за нерухомими об'єктами;
- інші об'єкти, що рухаються, затуляють вид на ціль.

1.2 Способи зображення об'єктів

В задачах спостереження об'єкт може бути визначений як будь-що важливе для подальшого аналізу. Об'єкти представлені їх виглядом та формою [2]. Способи представлення, що широко використовуються для відстеження, зображено на рис. 1.2:

1. об'єкти представляють собою точки, або набір точок (рис. 1.2(b)). Підходить для стеження за невеликими областями на зображенні;
2. форма об'єкта представлена прямокутником, еліпсом тощо (рис. 1.2(c, d)). Підходить для відстежування твердих об'єктів;
3. контур представляє собою границі об'єкта (рис. 1.2(g, h)). Силует знаходиться всередині контуру (Рисунок 1.2(i)). Підходить для стеження за нетвердими об'єктами;
4. з'єднані об'єкти складаються з частин тіла, що тримаються разом за рахунок суглобів. Наприклад, тіло людини - це з'єднаний об'єкт, що складається з ніг, рук, голови, торсу та суглобів (рис. 1.2(e)). Відношення між частинами регулюється моделями кінематичного руху;
5. скелетна модель – одна з найпоширеніших моделей для розпізнавання об'єктів.

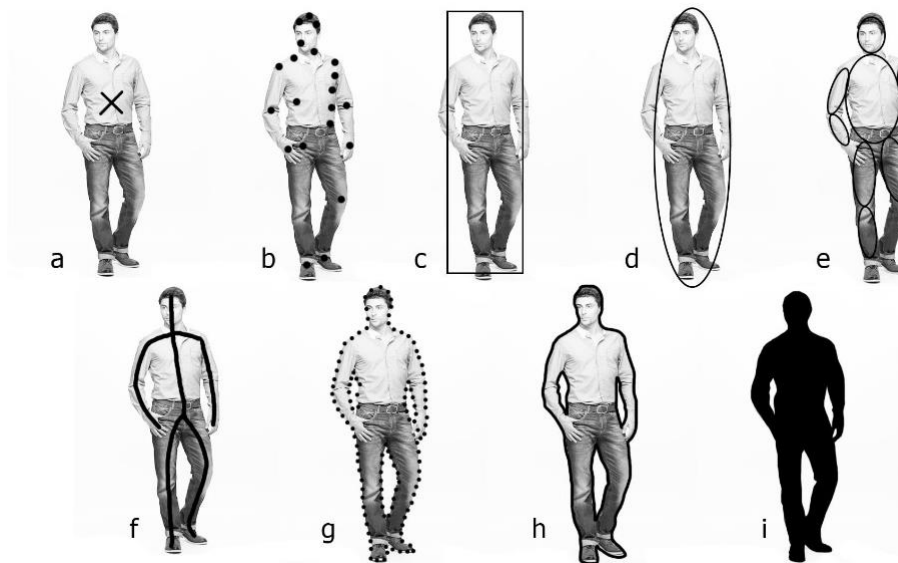


Рисунок 1.2 – Способи виділення об’єкта: а – центр об’єкта, b – особливі точки, c – форма об’єкта у вигляді прямокутника, d – об’єкт у формі еліпса, e – поєднання фігур, f – скелетна модель, g та h – контур об’єкта, i – силует

Також є ще кілька способів виділити об’єкта. Найбільш широко використовуються такі:

- щільність ймовірностей вигляду об’єкта може бути параметризована, наприклад, за Гаусом або змішування за Гаусом, чи непараметризована, як вікно Парзена;
- шаблони формуються з примітивних силуетів та форм. Їх перевага в тому, що вони несуть як просторову інформацію, так і про зовнішній вигляд. Використовуються, якщо положення об’єкта не змінюється;
- активна модель вигляду. Генерується одночасно форма і вигляд об’єкта. В основному, об’єкт представляє собою набір орієнтирів. Для кожного з орієнтирів зберігається вектор вигляду: текстура, колір, градієнт;
- багаторакурсна модель описує об’єкт з різних ракурсів.

1.3 Виділення характерних рис для відстежування

Виділення правильних рис грає критичну роль у відстежуванні об'єктів. В загальному випадку, характерні візуальні риси мають бути унікальними, щоб можна було виділити об'єкт відстежування порівняно з іншими. Виділення характерних рис тісно пов'язано з представленням об'єкта [9]. Так границі об'єкта є характерними рисами для представлення контуром. Основні візуальні риси:

- видимий колір, може бути представлений в RGB чи HSV (Hue, Saturation, Value);
- кордони об'єкта створюють значні зміни в інтенсивності зображення. Розпізнавання границь використовуються для знаходження цих змін. Важлива властивість границь – вони менш чутливі до змін світла порівняно з рисами кольору. Найбільш популярний є детектор Canny;
- оптичний потік – це представлення видимого сліду руху об'єктів, поверхонь, і граней візуальної сцени, що спостерігається під час відносного руху між спостерігачем (наприклад, око людини або камера) і сцени. Популярні алгоритми знаходження оптичного потоку: Хорна, Лукас-Канаде;
- текстура – це міра зміни інтенсивності поверхні, яка визначає такі характеристики як рівність та постійність.

1.4 Оптичний потік і його алгоритми

Дослідження оптичного потоку широко проводяться в областях аналізу руху. Процес виділення динамічних об'єктів складається з трьох основних етапів: попередня стабілізація параметрів зображення, виділення рухомих точок або областей, співставлення рухомих точок і областей між кадрами. Виділення рухомих точок і областей відбувається алгоритмами знаходження руху. Для розрахунку оптичного потоку (ОП) найбільш часто використовуються локальні і глобальні диференційні алгоритми.

Локальні диференційні алгоритми дозволяють достатньо швидко вирахувати оптичний потік, але точність отриманого ОП в деяких випадках є недостатньою для стабільного виділення рухомих об'єктів.

Глобальні методи розрахунку ОП є більш точними, але їх розрахункова складність не дозволяє використовувати їх в деяких задачах реального часу.

Алгоритми оптичного потоку не тільки визначають поле потоку, але і використовують оптичний потік при аналізі трьохвимірної сутності і структури сцени, а також 3D-руху об'єктів і спостерігача відносно сцени. Крім того, оптичний потік використовується для вивчення структури об'єктів.

Оптичний потік використовується в робототехніці при розпізнаванні об'єктів, слідкуванні за об'єктами, визначенні руху та при навігації робота.

Послідовності впорядкованих зображень дозволяють оцінювати рух як миттєву швидкість зображення, або як дискретне зміщення.

Задача розрахунку оптичного потоку полягає в тому, щоб оцінити зміщення зображення в кожній точці по даним про яскравість пікселів на послідовних кадрах. Серед широко відомих методів розрахунку можна виділити 3 основні групи: диференційні методи, розрахунок фазової кореляції та методи, оснований на співставленні окрестностей.

Основою для диференційних методів склали роботи Хорна (Horn) і Шунка (Schunk), а також Лукаса (Bruce Lucas) і Канаде (Takeo Kanade).

Алгоритм Лукаса – Канаде

Алгоритм Лукаса - Канаде широко використовується в комп'ютерному баченні. Основне рівняння оптичного потоку містить дві невідомі і не може бути однозначно вирішено. Алгоритм Лукаса-Канаде обходить неоднозначність за рахунок використання інформації про сусідні пікселі в кожній точці.

Метод заснований на припущенні, що в локальній області кожного

пікселя значення оптичного потоку однаково, таким чином можна записати основне рівняння оптичного потоку для всіх пікселів області і вирішити отриману систему рівнянь методом найменших квадратів.

Алгоритм Лукаса - Канаді менш чутливий до шуму на зображеннях, ніж поточечні методи, однак є суто локальним і не може визначити напрямку руху пікселів всередині однорідних областей.

Даний підхід називають диференційним, оскільки в його основі закладено розрахунок часткової похідної по горизонтальному і вертикальному напрямі зображення.

Припустимо, що $D=\{q_1, q_2, \dots, q_n\}$ – набір точок в межах точки $p(x, y)$. Тоді, отримаємо систему рівнянь 1.1:

$$\begin{cases} I_x(q_1)v_x + I_y(q_1)v_y = -I_t(q_1), \\ I_x(q_2)v_x + I_y(q_2)v_y = -I_t(q_2), \\ \dots \\ I_x(q_n)v_x + I_y(q_n)v_y = -I_t(q_n). \end{cases} \quad (1.1)$$

Зазвичай набір точок в околиці точки p обмежений вікном $N \times M$ з центром p . Проблема полягає в тому, що чим більша кількість точок в околиці точки p , тим більше рівнянь буде містити система для знаходження вектора \vec{v} . Наприклад, для вікна 3×3 потрібно вирішити систему з 9. Для вирішення даної проблеми використовується зважений метод найменших квадратів. Для визначення вагових коефіцієнтів для пікселей на зображенні використовується функція $W(x, y)$. Згідно зваженому методу найменших квадратів, для знаходження рішень необхідно мінімізувати відхилення:

$$\begin{aligned} \epsilon(v) &= \sum_{x,y \in D} W(x, y) [I(x, y, t) - I(x + \delta x, y + \delta y, t + \delta t)]^2 \\ &= \sum_{x,y \in D} W(x, y) \left(\frac{\partial I}{\partial x} v_x + \frac{\partial I}{\partial y} v_y + \frac{\partial I}{\partial t} \right)^2, \end{aligned} \quad (1.2)$$

де $v = (v_x, v_y)$.

Тоді,

$$\begin{aligned} \frac{\partial \epsilon(v)}{\partial v_x} &= 2 \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial x} v_x + \frac{\partial I}{\partial y} v_y + \right. \\ \left. \frac{\partial I}{\partial t} \right) \frac{\partial I}{\partial x} &= 2 \sum_{x,y \in D} W(x,y) \left[\left(\frac{\partial I}{\partial x} \right)^2 v_x + \frac{\partial I}{\partial y} \frac{\partial I}{\partial x} v_y + \frac{\partial I}{\partial t} \frac{\partial I}{\partial x} \right], \end{aligned} \quad (1.3)$$

$$\begin{aligned} \frac{\partial \epsilon(v)}{\partial v_y} &= 2 \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial x} v_x + \frac{\partial I}{\partial y} v_y + \right. \\ \left. \frac{\partial I}{\partial t} \right) \frac{\partial I}{\partial y} &= 2 \sum_{x,y \in D} W(x,y) \left[\frac{\partial I}{\partial x} \frac{\partial I}{\partial y} v_x + \left(\frac{\partial I}{\partial y} \right)^2 v_y + \frac{\partial I}{\partial t} \frac{\partial I}{\partial y} \right]. \end{aligned} \quad (1.4)$$

Для знаходження мінімуму похибки $\epsilon(v)$ необхідно прирівняти $\frac{\partial \epsilon(v)}{\partial v_x}, \frac{\partial \epsilon(v)}{\partial v_y}$ нулю. Отже, отримаємо:

$$\begin{cases} \sum_{x,y \in D} W(x,y) \left[\left(\frac{\partial I}{\partial x} \right)^2 v_x + \frac{\partial I}{\partial y} \frac{\partial I}{\partial x} v_y + \frac{\partial I}{\partial t} \frac{\partial I}{\partial x} \right] = 0, \\ \sum_{x,y \in D} W(x,y) \left[\frac{\partial I}{\partial x} \frac{\partial I}{\partial y} v_x + \left(\frac{\partial I}{\partial y} \right)^2 v_y + \frac{\partial I}{\partial t} \frac{\partial I}{\partial y} \right] = 0. \end{cases} \quad (1.5)$$

Рівняння можуть бути представлені в матричній формі:

$$\begin{aligned} &\begin{bmatrix} \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial x} \right)^2 & \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial y} \frac{\partial I}{\partial x} \right) \\ \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \right) & \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial y} \right)^2 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} + \\ &\begin{bmatrix} \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial t} \frac{\partial I}{\partial x} \right) \\ \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial t} \frac{\partial I}{\partial y} \right) \end{bmatrix} = 0. \end{aligned} \quad (1.6)$$

Нехай:

$$\begin{aligned} A &= \begin{bmatrix} \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial x} \right)^2 & \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial y} \frac{\partial I}{\partial x} \right) \\ \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \right) & \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial y} \right)^2 \end{bmatrix}, v = \begin{bmatrix} v_x \\ v_y \end{bmatrix}, B = \\ &\begin{bmatrix} \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial t} \frac{\partial I}{\partial x} \right) \\ \sum_{x,y \in D} W(x,y) \left(\frac{\partial I}{\partial t} \frac{\partial I}{\partial y} \right) \end{bmatrix}. \end{aligned}$$

Тоді,

$$A \cdot v + B = 0 \quad \Leftrightarrow = -A^{-1}B \quad (1.7)$$



Рисунок 1.3 – Приклад визначення оптичного потоку методом Лукаса-Канаде

Алгоритм Фарнебака

Farneback запропонував апроксимувати зміни інтенсивності в області за допомогою квадратичної форми:

$$I = xAx + bx + c. \quad (1.8)$$

З симетричною матрицею A (по суті, розглядаючи розклад по Тейлору до першого члена, ми брали лінійну апроксимацію $I = bx + c$, тобто зараз ми вирішили підвищити точність наближення). Якщо зображення змістилося в межах цієї області, то:

$$I_2(x) = I_1(x - d). \quad (1.9)$$

Підставляємо в квадратичний розклад, розкриваємо дужки, отримаємо:

$$A_2 = A_1, \quad (1.10)$$

$$b_2 = b_1 - 2A_1d, \quad (1.11)$$

$$c_2 = d^T A_1 d - b_1^T d + c_1. \quad (1.12)$$

Тепер ми можемо обчислити значення A , b , c на обох зображеннях, і тоді ця система стане надлишковою відносно d , і взагалі d можна отримати з другого рівняння:

$$d = -\frac{1}{2} A^{-1}(b_2 - b_1). \quad (1.13)$$

Доводиться виконувати наступну апроксимацію:

$$A = \frac{1}{2}(A_1 + A_2). \quad (1.14)$$

Позначимо для спрощення:

$$\Delta b = -\frac{1}{2}(d_2 - d_1). \quad (1.15)$$

Для компенсації шумів при обчисленні, знову звернемося до того припущення, що в області досліджуваної точки у всіх точок більш менш однакове зміщення. Тому знову проігноруємо похибку $|Ad - \Delta b|^2$ по вікну з гаусівськими ваговими коефіцієнтами w , і знайдемо вектор d , мінімізуючий цю сумарну похибку. Тоді ми отримаємо оптимальне значення:

$$d = (\sum \omega A^T A)^{-1} \sum \omega A^T \Delta. \quad (1.16)$$

І відповідну мінімальну похибку:

$$e = (\sum w \Delta b^T \Delta b) - d^T \sum w A^T \Delta.$$

Тобто нам потрібно для кожної точки розрахувати $A^T A$, $A^T \Delta b$, $\Delta b^T \Delta b$, усереднити по вікну, інвертувати матрицю і отримати результат. Відповідно ці множники можна розрахувати для всієї картинки і використовувати заздалегіть розраховані значення для різних точок.

Як звичайно, у цього алгоритма є деяка кількість модифікацій і покращень, в першу чергу дозволяючи використовувати відому апріорну інформацію – задану початкову апроксимацію потоку – і, знову таки, multi-scaling.

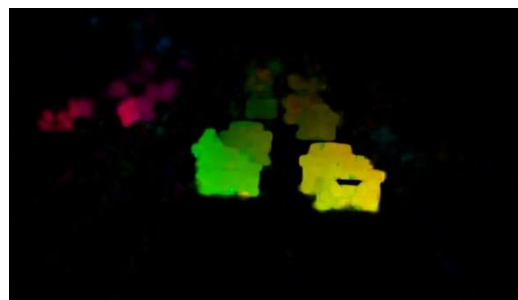
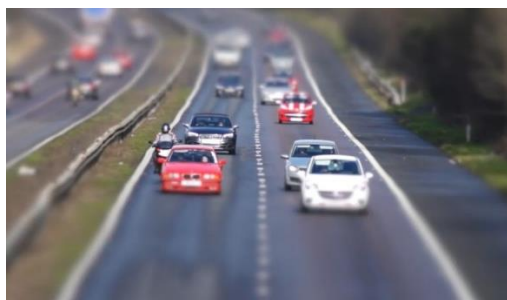


Рисунок 1.4 – Приклад визначення оптичного потоку методом Farneback

Алгоритм Horn–Schunck

Алгоритм Horn–Schunck, запропонований в 1981 році, оцінює оптичний потік, використовуючи поняття глобальної функції енергії:

$$E = \iint [(I_x u + I_y v + I_t)^2 + \alpha^2 (|\nabla u|^2 + |\nabla v|^2)] dx dy. \quad (1.17)$$

Оптичний потік визначається як векторне поле $v(r)$, мінімізуюче функціонал. Екстремум енергії можна знайти, вирішивши рівняння Ейлера-Лагранжа.

Метод Horn–Schunck носить більш глобальний характер, ніж метод Лукаса-Канаде. Він базується на тому, що на всьому зображенні оптичний потік буде досить гладким. Від того самого рівняння:

$$I_x dx + I_y dy + I_t dt = 0. \quad (1.18)$$

Пропонується перейти до функціоналу:

$$(I_x dx + I_y dy + I_t dt)^2 + \alpha^2 (|\partial u|^2 + |\partial v|^2). \quad (1.19)$$

Тобто додати вимоги на відсутність різкої зміни зміщень з ваговим коефіцієнтом α . Мінімізація цього функціоналу приводить нас до системи з двох рівнянь:

$$I_x^2 u + I_x I_y v = \alpha^2 \Delta u - I_x I_t, \quad (1.20)$$

$$I_x I_y u + I_y^2 v = \alpha^2 \Delta v - I_y I_t. \quad (1.21)$$

В цих рівняннях лапласіан пропонують розраховувати наближено:

$\Delta u = \langle u \rangle - u$; — різниця з середнім значенням. Отримаємо систему рівнянь, яку записуємо для кожного пікселя і вирішуємо загальну систему ітеративно:

$$u_{n+1} = u_n - I_x (I_x u_n + I_y v_n + I_t) / (I_x^2 + I_y^2 + \alpha^2), \quad (1.22)$$

$$v_{n+1} = v_n - I_y (I_x u_n + I_y v_n + I_t) / (I_x^2 + I_y^2 + \alpha^2). \quad (1.23)$$

В даному алгоритмі також пропонують використовувати multi-scaling, причому масштабувати зображення не в 2 рази, а з коефіцієнтом 0.65. Для великих зображень (більше 105 пікселів) алгоритм Horn–Schunck дуже складний.

Його часова складність має порядок $\Theta(SN)$, де S — кількість пікселів у зображенні, N — кількість ітерацій метода Якобі або Гауса–Зейделя.

Хоча проблема знаходження оптичного потоку вивчається вже декілька десятиліть, методи обробки ОП все ще продовжують покращуватися. Робота продовжується також через те, що при близькому розгляді проблема являється дуже непростою, а від якості визначення зміщень в обробці відео і зображень залежить стійкість і ефективність багатьох інших алгоритмів.



Рисунок 1.5 – Приклад визначення оптичного потоку методом Хорна-Шунка

1.5 Методи виявлення об'єктів

Виявлення об'єктів означає побудову деякого набору правил, яким повинен відповідати фрагмент зображення, для того, щоб бути визначеним. Також сюди можна віднести виявлення за допомогою шаблонів, заданих розробником. Виявлення об'єктів за допомогою шаблону полягає в перевірці кожної з областей зображення на відповідність заданому шаблону. З усього різноманіття існуючих алгоритмів виявлення об'єктів можна виділити декілька актуальних методів, які заслуговують уваги. Розглянемо особливості, переваги і недоліки кожного з них. З різноманіття існуючих алгоритмів виявлення об'єктів є кілька відповідних методів, які заслуговують на увагу. Розглянемо особливості, переваги та недоліки кожного.

Алгоритм Віола-Джонса названий на честь двох дослідників комп'ютерного зору, які запропонували метод у 2001 році: Пола Віола та Майкла Джонса.

Вони розробили загальну структуру виявлення об'єктів, яка змогла забезпечити конкурентоспроможні показники виявлення об'єктів у режимі реального часу. Він може використовуватися для вирішення різноманітних проблем виявлення, але відпочатку алгоритм розроблявся для виявлення облич.

Алгоритм Віола-Джонса має 4 основні етапи:

- вибір функцій, схожих на Хаар;
- створення цілісного образу;
- запуск тренувань AdaBoost;
- створення каскадів класифікаторів.

Надаючи зображення, алгоритм розглядає багато менших субрегіонів і намагається знайти обличчя, шукаючи конкретні особливості в кожному субрегіоні. Для цього потрібно перевірити багато різних положень і масштабів, оскільки зображення може містити безліч граней різного розміру. Віола та Джонс використовували функції Хаар для виявлення облич. Основні принципи, на яких заснований метод:

1. інтегральне представлення зображення. Це дає змогу швидко обчислювати необхідні об'єкти. Інтегральне представлення зображення – матриця, яка співпадає по розмірам з розмірами вхідного зображення в пікселях. Кожний елемент матриці представляє собою суму інтенсивностей всіх пікселів, які знаходяться лівіше і вище даного елемента;
2. використовуються ознаки, подібні до ознак Хаара, за допомогою яких виконується пошук потрібного об'єкта;
3. вибір ознак здійснюється з використанням алгоритму бустингу;
4. послідовної побудови композиції алгоритмів машинного навчання, коли кожний наступний алгоритм намагається компенсувати недоліки композиції всіх попередніх алгоритмів;
5. класифікатор представляє собою функцію, яка визначає належність деякого досліджуваного об'єкта до певного класу. Класифікатор, який

отримує на вхід ознаки, дає відповідь «істина» чи «брехня», дозволяє здійснювати виявлення;

6. в процесі виявлення використовуються каскади ознак. В основі такої ідеї лежить побудова послідовності класифікаторів, яка називається каскадом: кожний наступний прагне врахувати помилки попереднього;
7. для зображень використовується метод на основі скануючого вікна: виконується сканування зображень вікном пошуку, після чого до кожного положення застосовується класифікатор.

Переваги:

1. алгоритм являється найпопулярнішим і широко розповсюдженим методом виявлення об'єктів;
2. висока швидкість виявлення об'єктів за рахунок використання каскадного класифікатора;
3. висока точність знаходження з дуже низьким відсотком помилкових спрацювань, порівняно з точністю виявлення у повільніших алгоритмах;

Недоліки:

1. необхідна велика навчальна вибірка і багато часу на навчання;
2. обмеження на положення об'єкта при знаходженні [19].

Алгоритм порівняння з шаблоном. Порівняння шаблонів - це техніка пошуку ділянок зображення, схожих на патч (шаблон). Його застосування може бути робототехнікою або виробництвом. Патч - це невелике зображення з певними особливостями. Мета відповідності шаблонів - знайти патч / шаблон у зображенні. Відповідність шаблонів не є інваріантною за масштабами, а також не є інваріантною за поворотом.

Це дуже простий метод, де ми знаходимо найбільш співвідносну область. Для вводу без зміни масштабу та обертання цей метод чудово працює. Основний принцип цього методу полягає в тому, що б порівнювати зображення, яке обробляється з наявними шаблоном-зображенням. Для

роботи алгоритму необхідно мати вихідне зображення і шаблон.

Процес відбувається наступним чином:

- зображення, яке обробляється накладається на шаблон;
- зображення переміщається по шаблону в пошуках збігів;

при кожній зміні положення зображення обчислюється метрика, що відображає збіг. Метрика записується в підсумкову матрицю для кожного положення в вигляді;

По закінченню порівняння співпадіння знаходяться в глобальних максимумах або мінімумах, в залежності від обраного методу. Існують методи порівняння з шаблоном використовуючи методи найменших квадратів, кореляції та ін. Даний метод є найпростішим алгоритмом розпізнавання зображення, але має досить низьку точність.



Рисунок 1.6 – Пошук об'єкта методом порівняння з шаблоном

Сегментація зображень грає важливу роль в системах комп'ютерного зору для вирішення завдань, пов'язаних з розпізнаванням сцен і виділення (визначення) об'єктів. Існують певні вимоги, що пред'являються до оброблюваних областях:

- області повинні бути однорідними щодо заданих характеристик, тобто

внутрішня частина області повинна бути простою, не має великого числа отворів, що не відносяться до даної області;

- суміжні області повинні мати суттєві відмінності за характеристиками, щодо яких вони вважаються однорідними;
- межі між сегментами бути досить явними.

Сегментація є важливою процедурою, так як результати виконання даної процедури значно впливають на подальший процес аналізу зображення [11].

Нейромережіві методи включають в себе цілий клас алгоритмів. Основна ідея – послідовність перетворень сигналу паралельно працюючими функціональними елементами, нейронами. Сутність процесу навчання таких мереж зводиться до зменшення середньоквадратичної помилки. Системи виявлення об'єктів на зображенні, засновані на нейронних мережах, використовують ієрархічну структуру. Спочатку вектор ознак оброблюється грубою мережею з високим рівнем помилок другого роду, далі, якщо вектор не був класифікованим як не об'єкт, рішення корегується більш точною та повільнішою мережею. Нейронна мережа складається з декількох різних шарів, таких як вхідний шар, принаймні один прихований шар і вихідний шар. Вони найкраще використовуються при виявленні об'єктів для розпізнавання візерунків, таких як краї (вертикальні / горизонтальні), форми, кольори та текстури. Приховані шари - це звивисті шари в нейронній мережі цього типу, які діють як фільтр, який першим отримує вхід, перетворюють його за допомогою певного шаблону / функції та передають його наступному шару. З більш звивистими шарами щоразу, коли новий вхід надсилається наступному згортковому шару, вони змінюються по-різному. Наприклад, у першому згортковому шарі фільтр може ідентифікувати форму / колір у регіоні, а наступний може виявити об'єкт, яким він є насправді і останній згортковий шар може класифікувати об'єкт як шуканий об'єкт. В основному, чим більше і більше шарів входить, тим складніші шаблони можуть виявити нейронні мережі.

Для розпізнавання складних об'єктів створюють системи на основі нейронних мереж (НМ, neural network).

Вони можуть мати топологію, орієнтовану на розв'язання конкретної задачі із врахуванням властивостей об'єкта – просторово-часову орієнтацію, масштаб, геометричні параметри об'єкта, включаючи координати, кутове положення, лінійний розмір, відстань, тощо. У той же час істотним недоліком типових НМ є відсутність ефективних засобів для розв'язання задач розпізнавання динамічних образів. Основною проблемою інтерпретації динамічних візуальних сцен є висока розмірність простору ознак, наявність геометричних перетворень над об'єктом. Стиск простору ознак виконують методом витягу інтегральних і інваріантних до геометричних перетворень параметрів зображень. Метод геометричних та більш загальних алгебраїчних інваріантів відіграє значну роль у розв'язанні задач розпізнавання зображень. Так, наприклад, інваріанти, у тому числі інваріантні моменти, були успішно використані для розпізнавання профілів літаків і танків, друкованих і рукописних букв, параметрів стикувального вузла космічного апарата, а також багатьох інших об'єктів. Математичне обґрунтування інваріантних особливостей напівтонових зображень базується на теорії алгебраїчних інваріантів.

Суть НМ полягає в тому, що мережа складається з елементів, котрі називаються формальними нейронами (formal neuron). Кожен нейрон приймає набір сигналів, що надходять на його входи від одної групи таких же нейронів, обробляє сигнали з врахуванням попередніх сигналів і адаптації до них на основі процедур навчання і передає результати обробки другій групі нейронів. Зв'язки між нейронами кодуються вагами, що відображають важливість їх інформації для визначення загального результату.

Основний принцип настроювання нейронної мережі полягає в застосуванні процедур оптимізації та адаптації на основі певних критеріїв, здатності до перенавчання. Однією з переваг НМ є те, що всі елементи

можуть функціонувати паралельно, тим самим істотно підвищуючи ефективність розв'язання задач, особливо при обробці зображень в реальному часі.

Системи розпізнавання об'єктів зображення, що засновані на нейронних мережах, використовують ієрархічну архітектуру. Спочатку вектор ознак обробляється грубою з високим рівнем похибок, але швидкою, мережею, далі, якщо вектор не був класифікований як не об'єкт, алгоритм розв'язання коректується більш точною і більш повільною мережею.

Переважає кількість прикладних нейронних систем передбачає використання багат шарових персептронів (назва „персептрон” походить з англійського *perceptron* – сприйняття, оскільки перші зразки таких структур призначались для моделювання зору). Популярність персептронів зумовлена широким колом доступних для них задач [12].

Перевага нейронної мережі – це висока точність виявлення при авильному налаштуванні параметрів мережі.

Недоліки:

- чутливість до шуму;
- необхідність в ретельному налаштуванні параметрів нейронної мережі для отримання хороших результатів;
- схильність до повторного навчання;
- висока обчислювальна складність, і, як результат, швидкість роботи, недостатня для обробки в реальному часі;
- досить тривала процедура навчання. Багат шарові нейронні мережі.

Background subtraction

Знаходження переднього плану є однією з основних задач в області комп'ютерного зору і обробки зображень. Метою є знаходження змін в послідовності зображень. Віднімання фону – це будь-який метод, який дозволяє вилучити передній план зображення для подальшої обробки.



Рисунок 1.7 – Визначення об'єктів методом віднімання заднього фону (*Background subtraction*)

Висновки за розділом

В даному розділі розглянуто поняття відеоаналітики, проблеми пов'язані з розпізнаванням об'єктів та існуючі алгоритми та методи розпізнавання.

Проаналізувавши існуючі методи розпізнавання, можна зробити висновок, що для розпізнавання об'єктів в реальному часі краще застосовувати метод Віоли-Джонса. Алгоритм являється найпопулярнішим і широко розповсюдженим методом виявлення об'єктів. Алгоритм володіє високою швидкістю виявлення об'єктів за рахунок використання каскадного класифікатора та високою точністю знаходження з дуже низьким відсотком помилкових спрацювань.

Отже, для покращення процесу розпізнавання, шляхом додавання предобробки кадрів та розпаралелювання задач отримання даних з камери та їх обробки, будемо використовувати метод Віола-Джонса.

2 ВІДСТЕЖЕННЯ ОБ'ЄКТІВ У ВІДЕОПОТОЦІ

2.1 Відстеження об'єктів у відеопотоці

Під завданням виявлення динамічних об'єктів розуміється завдання виявлення і виділення ділянок зображення в послідовності кадрів. Відповідно, під виявленням певного об'єкта розуміється вибір одного або декількох виявлених динамічних об'єктів, які мають деякі схожі ознаки із заданим об'єктом пошуку [7]. Ознаки вибираються згідно з алгоритмом. Трекінгом (від англ. Tracking-стеження) називається визначення місця розташування рухомого об'єкта (декількох об'єктів) в реальному часі за допомогою камери. Відстеженням (англ. tracking) називають визначення розташування рухомого об'єкта (декількох об'єктів) з часом за допомогою відеокамери. Алгоритм аналізує кадри відео і видає положення рухомих цільових об'єктів відносно кадру.

Основною проблемою відстеження є зіставлення положень цільового об'єкта на послідовних кадрах, особливо якщо об'єкт рухається швидко порівняно з частотою кадрів [8]. Таким чином, системи відстеження зазвичай використовують модель руху, котра описує, як може змінюватись зображення цільового об'єкта при різноманітних його рухах. Прикладами таких простих моделей руху є:

- відстеження плоских об'єктів, модель руху – двовимірне перетворення (афінне перетворення або гомографія) зображення об'єкта (наприклад, початкового кадру);
- коли цільовим є жорсткий тривимірний об'єкт, модель руху визначає вигляд залежно від його положення у просторі та орієнтації;
- для стиснення відео, ключові кадри (англ. key frames) розділяють на макроблоки (англ. macroblocks). Модель руху являє собою розрив ключових кадрів, де кожен макроблок перетворюється за допомогою вектора руху, отриманого з параметрів руху;

- зображення деформовного об'єкта може бути покрите сіткою (англ. mesh), рух об'єкта задають положенням вершин цієї сітки.

Головне завдання алгоритму відстеження – це послідовний аналіз кадрів відео для оцінення параметрів руху. Ці параметри характеризують положення цільового об'єкта [16].

Трекінг для однієї камери

Найчастіше робота модулів трекінгу нерозривно пов'язана з роботою детектора руху. Для побудови траєкторій переміщення ведеться послідовний аналіз кожного кадру, на якому присутні рухомі об'єкти. У загальному випадку в одному кадрі може бути присутнім кілька рухомих об'єктів, тому програму необхідно не тільки побудувати траєкторії, але і розрізнити об'єкти і їх переміщення.

Трекінг по двом кадрам

Найбільш проста реалізація трекінгу розглядає два кадри і будує траєкторії по ним. Спочатку відзначаються переміщення на поточному та попередньому кадрі, далі, аналізуючи швидкість, напрямки руху об'єктів, а також їх розміри, обчислюються ймовірності переходу об'єктів з однієї точки траєкторії попереднього кадру в іншу точку поточного. Найбільш ймовірні переміщення присвоюються кожному об'єкту і складаються в траєкторію.

Трекінг по декільком кадрам

Об'єкти в кадрі можуть переміщатися по-різному: їх траєкторії можуть перетинатися, вони можуть зникати і виникати знову (наприклад, якщо камера стежить за автомагістраллю, то автомобіль в кадрі може перекриватися іншим, а потім знову виїжджати через нього), кілька об'єктів можуть об'єднуватися або різко змінювати напрямок руху. У цих випадках завдання побудови точної траєкторії ускладнюється.

Для аналізу подібних складних переміщень метод побудови траєкторії за двома кадрам не підходить, він дає високу похибку. Для підвищення точності трекінгу частина виробників використовують

технологію аналізу послідовності кадрів і безперервного постобробки отриманих результатів.

Програма будує графи - аналізує переходи об'єктів з одного стану в інший. Щоб зрозуміти, якого об'єкту яке переміщення відповідає, також аналізуються швидкості і напрямку руху, положення, колірні характеристики. Як результат видається набір найбільш ймовірних рухів об'єктів, який утворює траєкторію. Різниця методів полягає в тому, що при обробці послідовності кадрів враховується як поточний стан об'єкта, так і історія його переходів, що дозволяє підвищити точність в складних ситуаціях перетину руху, зникнень і виникнень об'єкта.

Розглянуті алгоритми добре працюють зі сценами, в яких об'єкти рухаються окремо один від одного, але для руху в скупченнях вони неспроможні.

Кореляційні методи

Для аналізу масових переміщень використовуються кореляційні методи: оператору необхідно задати область кадру, переміщення якої буде будуватися, потім програма почне пошук цієї області на наступних кадрах, після чого видасть траєкторію.

Як пошукова області також може виступати будь-який рухомий об'єкт, на який зреагував детектор руху, або об'єкт певного типу, який виявив класифікатор ПО. Програма складає гистограму квітів виділеної області, зазначає особливі точки (характерні кути, відстані), потім відбувається їх пошук на наступних кадрах [18].

Головний недолік кореляційного методу – його висока ресурсомісткість, так як на первинний аналіз пошукового зразка (виділення квітів, побудова гістограми, визначення особливих точок) потрібно в десятки, а іноді в сотні разів більше обчислювальних потужностей, ніж вимагає метод, заснований на детекторі руху.

Крім того, кореляційний метод будує траєкторії тільки заданих для пошуку об'єктів. Методи трекінгу за двома або кількома кадрам будують

траєкторії всіх рухомих об'єктів, надаючи інструмент пошуку траєкторії по об'єкту або об'єкта по його траєкторії. Кореляційний метод не застосуємо для сцен з високою інтенсивністю руху.

Міжкамерний трекінг

Перший спосіб

Передбачає установку синхронізованих відеокамер, які спостерігають за пов'язаними областями. Об'єкт просто переходить з поля зору однієї камери в поле зору іншого, програма фіксує цей перехід і як би "підхоплює" об'єкт і видає траєкторію. Дуже важливо для точної відпрацювання цього методу провести калібрування камер системи, підібрати обладнання зі схожими характеристиками, адже об'єкт повинен бути ідентичний при переході від однієї сцени спостереження до іншої. Чим довше перехід об'єкта, чим більше рухомих об'єктів в кадрі, чим більше кількість камер, між якими він переміщається, тим вище ймовірність помилки в результатах межкамерного трекінгу.

Другий спосіб

Заснований на роботі інтерактивного пошуку за прикметами і не вимагає установки спеціального обладнання. Оператору необхідно позначити існуючі камери на плані в програмі, задати середній час переходу від однієї камери до іншої, вибрати з архіву зразок об'єкта, траєкторію якого буде будувати система, або створити його самостійно - задати пропорції, розмір, вказати колірні характеристики. Програма видасть всі об'єкти, схожі на пошуковий запит, а оператору необхідно вибрати потрібний. Далі здійснюється пошук даного зразка на сусідніх камерах системи, програма аналізує план території, визначає, коли обраний об'єкт міг досягти тієї чи іншої камери, і видає відповідні результати у вигляді набору треків – згрупованих кадрів об'єкта з однієї IP-каналу. Угрупування зображень відбувається на підставі нерозривного руху об'єкта в поле зору камери.

Оператор вибирає з поданих наборів результат, відповідний шуканого об'єкту, запускає подальший пошук. Таким чином, пошук відбувається

покроково, і його можна продовжувати до тих пір, поки об'єкт не зникне з поля зору всіх камер системи або поки оператор не отримає достатні дані. Побудова траєкторії можна зупинити на будь-якому етапі.

Цей спосіб вимагає набагато більше часу оператора і набагато більше уваги, але і точність отриманих результатів дуже висока. До переваг методу також можна віднести той факт, що камери в системі не повинні бути пов'язані і строго відкалібровані - покрокове побудова траєкторії і алгоритми пошуку за прикметами роблять пошук незалежним.

Перенесення траєкторії з відео на план

Тема багатокамерного відеоаналізу дуже цікава для розробників, потенціал для вдосконалення технологій ще дуже великий. Наприклад, сучасні технології межкамерного стеження дозволяють відзначати на планах траєкторії переходів від однієї камери до іншої, а також будувати шляхи переміщення об'єктів на плані в рамках однієї камери - переносити переміщення об'єкта з відео на план.

У загальному випадку такий відеоаналіз може проводитися не тільки для відстеження переміщень, а й для побудови на плані теплових карт інтенсивності руху, підрахунку відвідувачів, контролю входу в зону і тривалого перебування в ній.

Завдання перенесення траєкторії з відео на план складна по ряду причин:

1. необхідність зв'язати площині спостереження камери і плану - головна складність;
2. спотворення;
3. прив'язка поля зору камери до області на плані;
4. відділення об'єктів один від одного.

Основні алгоритми пошуку об'єктів включають наступні:

1. щільний оптичний потік (Dense optical flow). Ці алгоритми допомагають оцінити вектор руху кожного пікселя у відеокадрі;
2. рідкий оптичний потік (Sparse optical flow). Ці алгоритми, такі як

Kanade-Лукас-Томаші (KLT), відстежують розташування кількох точок на зображенні;

3. фільтр Калмана. Це дуже популярний алгоритм обробки сигналів, який використовується для прогнозування розташування рухомого об'єкта на основі попередньої інформації про рух;
4. Meanshift і Camshift. Це алгоритми для визначення максимумів функції щільності. Вони також використовуються для відстеження;
5. окремі об'єкти-трекери. У цьому класі трекерів перший кадр позначений за допомогою прямокутника для позначення місця об'єкта, який ми хочемо відстежувати. Потім об'єкт відстежується в наступних кадрах за допомогою алгоритму відстеження. У більшості застосувань у реальному житті ці трекери використовуються разом із детектором об'єктів;
6. алгоритми пошуку декількох об'єктів (Multiple object track). У випадках, коли у нас є швидкий детектор об'єктів, є сенс виявити кілька об'єктів у кожному кадрі, а потім запустити алгоритм пошуку, який ідентифікує, який прямокутник в одному кадрі відповідає прямокутнику в наступному кадрі. Такий алгоритм показаний на рис. 2.1.

Завданню відстеження рухомих об'єктів у відеопотоці присвячена велика кількість наукових робіт. Існуючі методи відстеження рухомих об'єктів в відеопотоці можна розділити на чотири основні класи: відстеження областей, відстеження по активному контуру, відстеження за характерними ознаками, відстеження за моделлю.



Рисунок 2.1 – Приклад відстеження багатьох об'єктів (Multiple object track).

Методи відстеження областей

Методи відстеження областей відстежують об'єкти по зміні областей кадру, відповідних рухомих об'єктам. В таких методах зображення заднього фону динамічно оновлюється, а області руху зазвичай виділяються шляхом віднімання фону з поточного кадру.

Методи відстеження областей успішно аналізують сцени, що містять невелику кількість рухомих об'єктів на постійному фоні. Однак, в умовах динамічно мінливого фону і взаємного перекриття рухомих об'єктів, дані методи мають низьку надійність і можуть використовуватися тільки в якості попередньої обробки відеопотоку для застосування методів інших класів.

Відстеження точок

Відстежування може бути сформульоване як визначення об'єктів по точках покадрово. Супроводження точок складна задача через наявність перекриттів, неправильних визначень, входження і виходу об'єкта. Методи розділяються на дві різні категорії: детерміністичні і статистичні. Детерміністичні методи для супроводження точок визначаються ціною асоціації кожного об'єкта на попередньому кадрі з одним об'єктом на теперішньому кадрі, використовуючи набір рухових обмежень.

Супроводження визначається комбінацією наступних обмежень:

- близькість;
- максимальна швидкість;
- незначні зміни швидкості (гладкий рух);
- загальні рухи;
- жорсткість.

Ці обмеження також можуть бути використані в контексті відстежування точок статистичними методами.

Методи відстеження по активному контуру

Методи відстеження по активному контуру відстежують об'єкти шляхом подання їх обрисів у вигляді обмежуючих контурів і динамічного оновлення цих контурів на наступних кадрах. Ці методи мають на меті пряме вилучення форми об'єктів і дають більш повний опис об'єктів в порівнянні з методами відстеження областей.

На відміну від методів відстеження областей, методи відстеження по активному контуру описують об'єкти більш просто і ефективно, а також скорочують обчислювальну складність. Вони можуть безперервно відстежувати об'єкти навіть в умовах спотворень і часткового перекриття об'єктів. Однак, методи відстеження по активному контуру дуже чутливі до точності початкової ініціалізації відстеження, що ускладнює запуск процесу відстеження в автоматичному режимі.

Методи відстеження за характерними ознаками

Методи відстеження за характерними ознаками виконують відстеження об'єктів шляхом вилучення характерних елементів кадру, оцінки їх кількісних параметрів і наступного порівняння з характерними ознаками інших кадрів. Методи відстеження за характерними ознаками можна розділити на три групи відповідно до природи використовуваних ознак: методи на основі глобальних ознак, методи на основі локальних ознак, методи на основі графів залежностей. До глобальних ознак відноситься центр ваги, а також розподіл яскравості та кольору кадру. До

локальних ознак належать прямолінійні і криволінійні сегменти, а також точки стику контурів областей кадру. Методи на основі графів залежностей використовують для відстеження відстані та геометричних відносин між характерними ознаками.

Методи відстеження за характерними ознаками можна легко і швидко адаптувати для обробки і відстеження декількох об'єктів в реальному часі.

Такі методи можуть справлятися з частковим перекриттям об'єктів, використовуючи інформацію про рух об'єкта, локальні ознаки і графи залежностей. Однак ці методи мають ряд серйозних обмежень. Точність відстеження об'єктів на основі двовимірних ознак зображень є невисокою через нелінійні спотворення при проектуванні і варіації зображення, при зміні точки огляду.

Методи відстеження по моделі

Методи відстеження по моделі відстежують об'єкти шляхом зіставлення ділянок зображення проєкцій тривимірних моделей цих об'єктів, складених по апіорним даним. Моделі зазвичай будуються попередньо за допомогою ручних вимірювань, інструментів автоматизації проектних робіт або методів комп'ютерного зору.

У порівнянні з іншими методами відстеження, методи відстеження по моделі мають наступні основні переваги.

По-перше, завдяки використанню апіорних знань про тривимірні контури або поверхні об'єктів, ці методи є стійкими. Вони можуть отримувати кращі результати навіть в умовах перекриття об'єктів і зближення кількох осередків руху на зображенні. По-друге, після встановлення геометричної відповідності між двовимірними координатами зображення і тривимірними просторовими координатами шляхом калібрування камери, метод природним чином отримує тривимірне положення об'єктів. По-третє, методи відстеження по моделі застосовні навіть у випадках, коли об'єкти сильно змінюють свою орієнтацію в процесі руху.

До недоліків методів даного класу можна віднести необхідність побудови моделей об'єктів, високу обчислювальну складність.

Порівняння методів відстеження.

В умовах взаємного перекриття об'єктів і невисокої точності початкової ініціалізації процесу відстеження, можливості відстеження по областям і по активному контуру обмежені.

Методи відстеження по моделі мають високу надійність, однак вони вимагають великих обчислювальних витрат і не підходять для вирішення задач в реальному часі.

Кожні методи відстеження потребують механізму визначення об'єкта на кадрах чи коли об'єкт вперше з'являється на відео. Загальний підхід у визначенні об'єктів – використання інформації з одиничного кадру. Але деякі алгоритми можуть використовувати тимчасову інформацію, яка була підрахована з послідовності кадрів.

2.2 Проблема перекриття при відстежуванні об'єктів

На проблемі перекриття об'єктів варто зупинитися докладніше. Дійсно, вихідний об'єкт може при своєму переміщенні частково або повністю зникнути з поля зору на невизначений час, а трекер повинен продовжувати відстеження цілі в звичайному режимі.

Перекриття дуже впливають на якість відстеження і можуть значно його погіршити. Найчастіше, ситуація перекриття повинна бути визначена до того, як ціль буде точно локалізована в кадрі шляхом маскування перекритою частиною, в той час, як визначити перекриту частину цілі можна буде тільки після того, як ціль буде локалізована в кадрі. Зробимо огляд існуючих варіантів розв'язання проблеми.

Адаптація до змін мети.

У книзі Jepson D., Fleet DJ, El-Maraghi Robust Online Appearance Models for Visual Tracking пропонується алгоритм адаптивного трекінгу, що використовує для представлення мети три компонента-розподілу:

стабільний S (той, який навчають весь період часу), перехідний W (за два кадри) і компонент викидів L. При цьому перекриття характеризуються компонентом викидів, які мають рівномірний розподіл. Алгоритм адаптує модель до повільних змін цілі, навчаючи таким чином стабільний компонент. Він досить надійний і використовується для оцінки руху, а перехідний - як додаткове обмеження. Обидва цих компонента мають розподіл Гаусса.

Навчання моделі проводиться за допомогою ЕМ-алгоритму, адаптованого під використання в реальному часі. При цьому важливо, що трекер може реалізовувати будь-які ознаки зображень. У цій книзі пропонується застосування комплексних коефіцієнтів спеціального смугового фільтра для обробки зображень керованої піраміди (steerable pyramid). Така модель забезпечує стабільність алгоритму при зміні масштабу і просторового оточення. Недоліки запропонованого алгоритму:

- не враховує кольоровість, працює тільки з повільно рухомими об'єктами;
- не справляється із завданням, якщо в оточенні є схожі об'єкти або відбувається повне перекриття.

Інший підхід визначає перекриття пікселів при перевищенні помилки деякого порога. Такі алгоритми добре працюють, якщо статистичні параметри перекриттів збігаються з очікуваними.

Однак, на жаль, в більшості випадків очікування не виконуються, так як в реальних ситуаціях перекриття можуть бути схожі на саму ціль або перекривати її протягом тривалого часу.

Виявлення перекриттів завдяки зміні руху окремих областей

У книзі Hariharakrishnan K., Schonfeld D. Fast object tracking using adaptive block matching представляється блоковий контурний трекер. У цьому алгоритмі ситуація перекриття аналізується шляхом порівняння параметрів руху між метою та блоками зображення, які не можуть бути компенсовані рухом (наступними кадрами). Ціль представляється маскою,

яка ініціалізується в кілька кроків: спочатку виконується багатозначна сегментація за чотирма напрямками, а потім - пошук області однаково рухомих сегментів (таким чином, аналізується просторово-часовий контекст об'єкта).

Алгоритм передбачає невелику інтенсивність руху в кадрі і аналізує тільки кожен третій кадр. В процесі роботи алгоритм шукає блоки зображення, що належать цілі, границям мети та задньому фону.

Далі робота йде з отриманими блоками. Рух між кадрами визначається шляхом застосування афінних перетворень до початкових блокам, якими ініціалізувалася ціль трекінгу. Відмінною особливістю алгоритму є розгляд подій перекриття і звільнення від перекриття, як взаємнозворотніх (для визначення звільнення від перекриття, поточний кадр послідовності компенсується рухом попереднього (наступного) кадру для отримання відкритих (перекритих) областей). У разі звільнення від перекриття, відкриті (перекриті) області, що володіють запропонованими характеристиками руху, ідентичні (розрізняються) з об'єктом. Алгоритм шукає такі подібності (відмінності) руху. Якщо в якійсь області рух припинився (або відрізняється від рухом цілі), значить, ця область перекрита. Область позначається неперекрытою, якщо вона відновлює рух.

Відновлення руху визначається шляхом кластеризації векторів руху областей цілі і обчисленням відстані від вектора руху розглянутої області до центру цього кластера. Якщо рух якийсь із областей відновилося, маска оновлюється.

Використання карти перекриттів

Алгоритм контентно-адаптивного прогресивного аналізу перекриттів (CAPOA).

На відміну від звичайних, що враховують тільки зовнішній вигляд цілі, даний алгоритм поетапно аналізує ситуацію перекриття, беручи до уваги просторово-часовий контекст. При цьому отримана інформація двічі перевіряється на відповідність цілі і обмеженням переміщення - це дозволяє

краще розділяти цільовий об'єкт з об'єктами-перекриттями. Для виключення ситуацій помилкового визначення положення цілі, при перекриттях використовується зіставлення з адаптованим шаблоном. Проблема перекриття вирішена завдяки порівнянню з адаптованим шаблоном (алгоритм VMTM), де неперекритих частина мети використовується для установки цілі з помилкового положення в правильне. В результаті трекер стає стійким до великої кількості типів перекриттів.

У початковий момент часу ціль вручну виділяється визначенням області інтересу (ROI). Коли приходить новий кадр, апроксимація ROI знаходиться з першого співпадіння за зразком. Однак місце розташування цілі, визначене по першому співпадінню, може бути помилковим, так як використовує шаблон, сформований відповідно до ситуації з перекриттями на попередньому кадрі. Пошук відповідності шаблону здійснюється перетворенням системи координат для відображення припущеного шаблону на кадр і знаходження області, що найкраще узгоджується з шаблоном. При цьому перекриття вже є повністю замаскованими завдяки SAPOA.

Алгоритм SAPOA отримує на вхід чотири аргументи: ROI, ціль, попередній кадр і попередню карту викидів. На виході алгоритм видає карту викидів і оновлену маску шаблону.

Визначення перекриттів відбувається на блоках, а не на окремих пікселях, так як просторовий контекст відіграє важливу роль у визначенні перекриттів (особливо, якщо перекриття має схожість з метою).

Для знаходження компромісу між надійністю і роздільною здатністю, використовується прогресивна процедура сканування. ROI сканується багато разів, і з кожним проходом розмір блоку зменшується вдвічі.

При цьому аналізуються тільки ті блоки, в яких ситуація з перекриттями не була вирішена на попередньому кроці. Сканування закінчується, коли ситуація з перекриттями в ROI стає повністю визначеною. Для того, щоб визначити перекриття, використовуються

попередні кадри – для кожного блоку шукається схожий на нього блок на попередньому кадрі.

Якщо це необхідно, пошук триває до тих пір, поки ситуація з перекриттями для даного блоку не буде точно встановлена (пошук може дійти до першого кадру, де ситуація чітко визначена для кожного блоку). Ситуація з перекриттями на даному кадрі задається бінарною картою викидів. При скануванні ROI зі старої карти викидів генерується нова, тимчасова. Якщо неможливо однозначно визначити, перекритий блок чи ні, то використовується оцінка руху для даного блоку.

Модель цілі на кожному кроці алгоритму при необхідності масштабується і адаптується за допомогою фільтра Калмана до змін у відстежуваному об'єкті. Основне завдання алгоритму VMTM полягає в підвищенні точності визначення місцеположення цілі. Це досягається пошуком локального мінімуму спеціальної функції, яка враховує попередню карту викидів. Таким чином, уникають перешкоди, пов'язані з можливими перекриттями. Даний метод дає в цілому хороші показники, однак він не справляється з повними перекриттями і об'єктами-перекриттями, ідентичними відстежуваному об'єкту.

Спільне використання трекара і детектора

TLD (Tracking-Learning-Detection)

Метод демонструє хороші показники і має комерційний успіх. Розроблений для довгострокового відстеження будь-якого класу цілей. На відміну від алгоритмів, що спеціалізуються тільки на відстеженні об'єкта або тільки на покадровому пошуку об'єкта, TLD використовує обидві ідеї і скріплює їх стадією навчання. TLD складається з трекара, аналізуючого рух цілі від кадру до кадру і детектора, незалежно скануючого кожен кадр послідовності.

При цьому детектор може робити помилки двох типів: помилкові спрацьовування і помилкові відмови.

Навчальний компонент відстежує стан і трекара, і детектора і генерує

навчальну вибірку для зменшення числа помилок останнього. При цьому навчальний компонент передбачає можливість одночасного виходу з ладу і трекера, і детектора. В процесі навчання детектор навчається визначати більш широкий діапазон варіацій представлення цілі і чіткіше розділяти ціль від заднього фону. Особливий інтерес представляє запропонований навчальний компонент (P-N Learning). Основна ідея полягає в використанні двох експертів: Р-експерта, який визначає тільки помилкові відмови, і N-експерта - помилкові спрацьовування.

Незважаючи на те, що самі експерти допускають помилки, їх незалежне використання компенсує ці помилки. Детектор включає в себе бінарний класифікатор, скануючий вікно і модель цілі, що складається з навчальної вибірки прикладів представлення цілі. Кожен елемент вибірки обробляється незалежно від інших. При наявності N вузлів в регулярній сітці сканування виходить $2N$ комбінацій для даного кадру. При цьому може бути ситуація, коли одній цілі відповідає відразу кілька місць розташування, що неможливо, так як порушується зв'язність руху. В цьому випадку виникає «помилкове спрацьовування».

Саме зв'язність руху є одним з головних параметрів для визначення помилок детектора. Р- експерт використовує тимчасову зв'язність і передбачає, що вона рухається по траєкторії.

Він застосовує результати роботи трекера для оцінки переміщення цілі і, якщо детектор відбраковує найбільш ймовірне (з точки зору трекера) місце розташування цілі, генерує відповідну навчальну вибірку. Вона генерується з поточної цілі за допомогою переміщення, масштабування і поворотів (близько 100 екземплярів).

N-експерт використовує просторову зв'язність мети і передбачає, що ціль перебуває тільки в єдиному місці.

Він також застосовує дані з трекера і серед результатів детектора знаходить найкращий, який оновлює стан трекера, а решту додає до вибірки, як негативні екземпляри. Як було згадано вище, в якості моделі цілі

використовується колекція негативних і позитивних екземплярів - областей зображення. Подібність між двома екземплярами розраховується як нормалізована кореляція. Вона ж і виступає критерієм в класифікаторі по найближчому сусіду (Nearest Neighbor).

Саме цей класифікатор приймає рішення про оновлення моделі. Детектор використовує скануюче вікно для пошуку відповідності за шаблоном. При великій кількості шаблонів пошук може зайняти дуже багато часу.

Тому використовується каскад з трьох класифікаторів: по різниці моделей, класифікатор ансамблів і по найближчому сусіду. На практиці до останнього класифікатора доходить близько 50 екземплярів, а таку кількість вже можна швидко обробити за допомогою NN-класифікатора. В якості трекара використовується Median-Flow трекер, який є розширенням трекара Лукаса - Канада (Lucas - Kanade), що працює з оптичними потоками. Розширення включає в себе визначення факту втрати мети: в цьому випадку трекер просто не повертає розташування (обмежувальну рамку).

Факт втрати мети встановлюється, якщо різниця зміщення деякої точки цілі і медіани всіх точок перевищує заданий поріг. Це може відбуватися при швидкому переміщенні або перекритті цілі. Якщо на певному етапі ні детектор, ні трекер не видає розташування цілі, вона вважається втраченою.

В іншому випадку вибирається найкращий результат (шляхом розрахунку кореляції з шаблоном).

При такому підході детектор спирається на вже відомі шаблони, але може знайти ціль, якщо вона різко змінила розміщення або вийшла з довгого перекриття, а трекер вносить нові шаблони, оновлюючи базис детектора, і враховує просторово-часовий контекст.

При всіх своїх перевагах, метод має такі недоліки:

- погана обробка ситуації поворотів цілі;
- навчання тільки детектора (трекер залишається статичним),

немає можливості відстеження декількох цілей;

– слабка підтримка рухомих каркасних об'єктів.

Найбільш цікавим і комплексним методом з розглянутих є TLD. Ідея паралельного застосування трекера, що реалізує принцип просторово-часової безперервності траєкторії цілі і детектора, який здійснює пошук за шаблоном, дозволить використовувати сильні і нівелювати слабкі сторони обох підходів. Застосування простого детектора із пошуком по всьому кадру природно для вирішення проблеми повного перекриття мети, з подальшим її виникненням в невідомих координатах.

У той же час, такий підхід не дозволяє легко адаптуватися до змін цілі і провокує велику кількість помилок (помилкових спрацьовувань і відмов). Застосування простого трекера, що відслідковує ціль по її траєкторії і використовує інформацію з попередніх кадрів, дозволяє адаптуватися до змін зовнішнього вигляду цілі і уникнути помилок, пов'язаних з помилковими спрацьовуваннями (якщо вибрані відповідні поставленому завданню ознаки), проте є абсолютно непридатним для боротьби з повними тривалими перекриттями. Тому, для побудови комплексного методу, що успішно вирішує проблему перекриттів, необхідно використовувати обидва підходи.

Висновки за розділом

В даному розділі розглянута задача пошуку об'єктів у відеопотоці та його відстежування. Під знаходженням певного об'єкта розуміється вибір одного або декількох виявлених динамічних об'єктів, які мають деякі схожі ознаки із заданим об'єктом пошуку. Ознаки вибираються згідно з алгоритмом. Трекінгом (від англ. Tracking-стеження) називається визначення місцярозташування рухомого об'єкта (декількох об'єктів) в часі за допомогою камери. Були розглянуті алгоритми та методи пошуку об'єктів відеопотоку, а саме фільтр Калмана, оптичний потік, Meanshift та Camshift.

Також у розділі висвітлена проблема перекриття одними об'єктами

інших. Об'єкти відстежування можуть частково або повністю перекриватися іншими об'єктами на деякий проміжок часу. Для вирішення цієї проблеми можна:

- зробити навчання каскада Хаара для підвищення точності розпізнавання при значному перекритті об'єктами;
- застосування багатокамерного трекінгу для співставлення об'єктів на різних кадрах одного моменту часу. Це потребує точної калібровки камери. Дозволяє позбавитись недоліків оклюзії одного об'єкта іншим.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПІДСИСТЕМИ

3.1 Інструментарій розробки

Метою роботи є підвищення якості процесу розпізнавання об'єктів відеопотоку шляхом використання операції предобробки кадрів та збільшення значення fps для інтелектуального робота в режимі реального часу. Програмний код створений в середовищі розробки PyCharm, на мові програмування Python, з використанням бібліотеки OpenCV рис. 3.1.

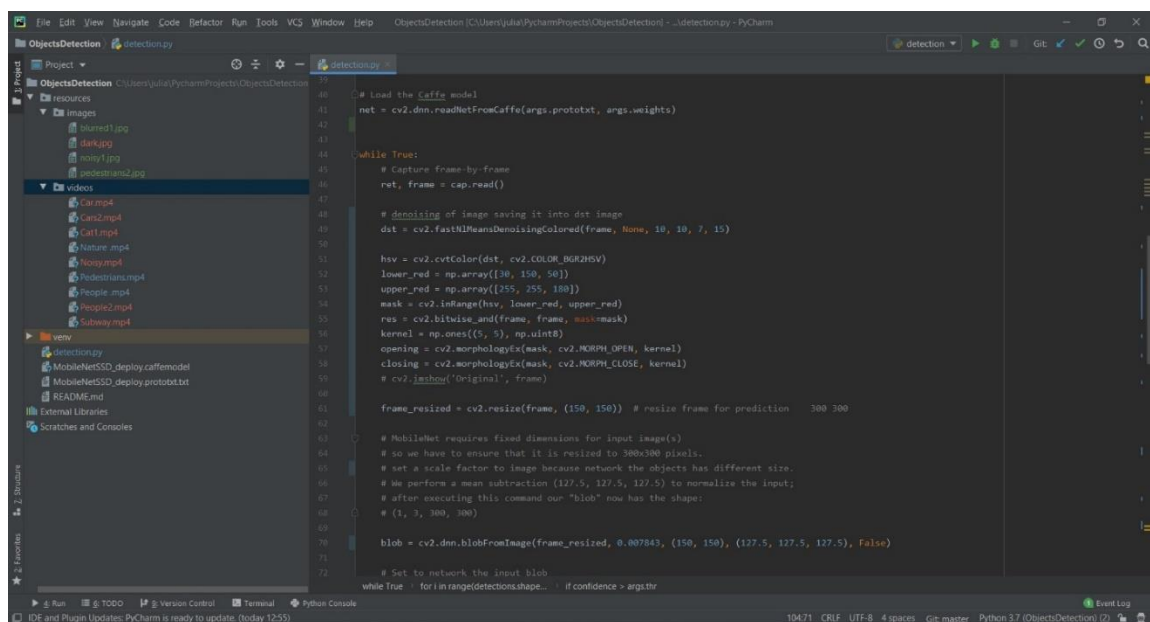


Рисунок 3.1 – Середовище розробки

Розглянемо програмні та операційні засоби, які будуть використовуватися при розробці програмного забезпечення:

Windows 10 – операційна система для персональних комп'ютерів і робочих станцій, розроблена корпорацією Microsoft в рамках сімейства Windows NT після Windows 8.1. Система покликана стати єдиною для різних пристроїв, таких як персональні комп'ютери, планшети, смартфони, консолі Xbox One і ін. Windows 10 поставляється в якості послуги з випуском оновлень протягом усього циклу підтримки.

Згідно зі статистичними даними сайту W3Schools, Windows 10 займає перше місце в світі серед операційних систем, що використовуються для доступу до мережі Інтернет, випередивши в квітні 2017 року попереднього лідера - Windows 7.

Python – мова програмування, яка має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис, динамічна обробка типів, а також те, що це інтерпретована мова, роблять його ідеальним для написання скриптів та швидкої розробки прикладних програм у багатьох галузях на більшості платформ. Python – це універсальна мова, що широко використовується в усьому світі для самих різних цілей – бази даних і оброблення текстів, вбудовування інтерпретатора в ігри, програмування GUI, швидке створення прототипів (RAD) програмування Internet і Web додатків – серверних (CGI), клієнтських (роботи), Webсерверів і серверів додатків. Серед переваг мови Python можна виділити переносимість написаних програм, на комп'ютери різної архітектури та з різними операційними системами, лаконічність запису алгоритмів, можливість отримати ефективний код програм за швидкістю виконання. Зручність мови Python ґрунтується на тому, що вона є мовою високого рівня, має набір конструкцій структурного програмування та підтримує модульність. Гнучкість та універсальність мови Python забезпечує її широке розповсюдження. Python – молода сценарна мова, історія якого почалася в 1990 році, коли співробітник голандського інституту CWI, тоді ще мало кому відомий Гвідо ван Росум приймав участь в проєкті створення мови ABC. Ця мова була призначена для заміни мови BASIC в навчанні студентів основних концепцій програмування. Паралельно з роботою над основним проєктом Гвідо ван Росум вдома на своєму Macintosh написав інтерпретатор іншої простої мови але деякі принципи мови ABC все ж були запозичені. На честь англійського колективу комічних акторів (яких дуже любляв Гвідо) "Monty Python's Flying Circus" було названо мову та почалося її розповсюдження мережею

Internet.

Мова почала швидко розвиватися, оскільки з'явилася велика кількість людей, що були зацікавлені та розумілися в розвитку мов програмування. Спочатку це була досить проста мова, невеликий інтерпретатор, незначна кількість функцій, об'єктноорієнтоване програмування було відсутнім, але дуже швидко все це з'явилося та до сьогодення продовжується її розвиток та виходять нові версії, де кожна наступна має декілька суттєвих відмінностей від попередньої. Інтерпретатори Python існують під всі можливі платформи: Windows, UNIX та ін. Всі вони розповсюджуються безкоштовно. Python є однією з десяти найпопулярніших мов програмування. Найбільша проблема, з якою можна наразі зіткнутися, – це вибір однієї з двох існуючих версій Python. Остання версія Python 2 має номер 2.7, вона ще довго буде підтримуватися, але версія Python 2.8 ніколи не вийде. Нова розробка буде вестися лише на Python 3. Python – інтерпретована мова програмування, що створює байт-код (файли з розширенням .py, які з'являються у папці із текстами програм під час їх виконання) для більш швидкої роботи.

PyCharm — інтегроване середовище розробки для мови програмування Python. Надає засоби для аналізу коду, графічний зневаджувач, інструмент для запуску юніт-тестів і підтримує веб-розробку на Django. PyCharm розроблена російською компанією JetBrains на основі IntelliJ IDEA. PyCharm працює під операційними системами Windows, Mac OS X і Linux. Можливості середовища розробки:

- статичний аналіз коду, підсвічування синтаксису і помилок;
- навігація серед проектів і сирцевого коду: відображення файлової структури проекту, швидкий перехід між файлами, класами, методами і використаннями методів;
- рефакторинг: перейменування, витяг методу, введення змінної, введення константи, підняття і опускання методу тощо;
- інструменти для веб-розробки з використанням фреймворку Django;

- вбудований зневаджувач для Python;
- вбудовані інструменти для юніт-тестування;
- розробка з використанням Google App Engine;
- підтримка систем контролю версій: загальний користувацький інтерфейс для Mercurial, Git, Subversion, Perforce і CVS з підтримкою списків змін та злиття.

PyCharm Professional Edition має кілька варіантів ліцензій, які відрізняються функціональністю, вартістю та умовами використання. PyCharm Professional Edition безкоштовна для освітніх установ і проектів з відкритим сирцевим кодом. Існує також вільна версія Community Edition з усіченим набором можливостей, яка поширюється під ліцензією Apache 2.

OpenCV (англ. Open Source Computer Vision Library) – бібліотека алгоритмів комп'ютерного зору, обробки зображень та чисельних алгоритмів загального призначення з відкритим кодом. Бібліотека реалізована на C / C ++, може використовуватися для Python, Java, Ruby, Matlab, Lua та інших мов. Бібліотека містить понад 2500 оптимізованих алгоритмів, серед яких повний набір як класичних так і практичних алгоритмів машинного навчання і комп'ютерного зору. Алгоритми OpenCV застосовують у таких сферах:

- аналіз та обробка зображень;
- системи з розпізнавання обличчя;
- ідентифікації об'єктів;
- розпізнавання жестів на відео;
- відстежування переміщення камери;
- побудова 3D моделей об'єктів;
- створення 3D хмар точок зі стерео камер;
- склеювання зображень між собою, для створення зображень всієї сцени з високою роздільною здатністю;
- система взаємодії людини з комп'ютером;

- пошуку схожих зображень із бази даних;
- усування ефекту червоних очей при фотозйомці зі спалахом;
- стеження за рухом очей;
- аналіз руху;
- ідентифікація об'єктів;
- сегментація зображення;
- трекінг відео.

NumPy – бібліотека обчислень для Python з відкритим вихідним кодом. Містить такі структури даних, як багатовимірні масиви та матриці. Може слугувати для виконання певних математичних операцій над масивами, на зразок тригонометричних, статистичних, алгебраїчних обчислень. Бібліотека містить багато математичних, алгебраїчних функцій та функцій перетворення [4].

3.2 Аналіз бібліотеки OpenCV

OpenCV (Open Source Computer Vision Library) випущений під ліцензією BSD і, отже, безкоштовна як для академічного, так і для комерційного використання. Вона має інтерфейси для мов C++, Python і Java і підтримує Windows, Linux, Mac OS, iOS та Android. Бібліотека OpenCV була розроблена для задач розпізнавання. Мова на якій вона написана це C/C++, сама ж бібліотека може скористатися багатоядерною обробкою. Бібліотека використовує OpenCL, вона може скористатися апаратним прискоренням базової неоднорідної обчислювальної платформи. Прийнята по всьому світу, OpenCV налічує понад 47 тисяч користувачів спільноти та приблизну кількість завантажень понад 14 мільйонів. OpenCV – це бібліотека з набором методів, алгоритмів та технологій для роботи із розпізнаванням зображень. Бібліотека має в собі готові методи для використання користувачем, тому написати просту програму для розпізнавання не завдає великих труднощів. Сама в собі бібліотека має

готові каскади, що дають змогу знаходити певні нам об'єкти.

OpenCV містить в собі такі алгоритми: інтерпретація зображення, калібровка камери, усунення оптичних шумів, визначення подібності, аналіз переміщення об'єкта, сегментація зображення, аналіз жестів.

Основні модулі можна віднести до 4 груп:

1. модулі `core`, `highgui`, які реалізують базову функціональність (базові структури, математичні функції, лінійна алгебра, ввід/вивід зображення);
2. модулі `imgproc`, `features2d` для обробки зображення (фільтрації, геометричні перетворення, сегментація, пошук особливих точок);
3. модулі `video`, `objdetect`, `calib3d` (калібровка камери, аналіз руху, пошук положення в просторі, побудова карта глибин, оптичний потік);
4. модуль `ml`, який реалізує алгоритми машинного навчання (метод ближніх сусідів, наївний байесівський класифікатор, машина опорних векторів, нейронні мережі).

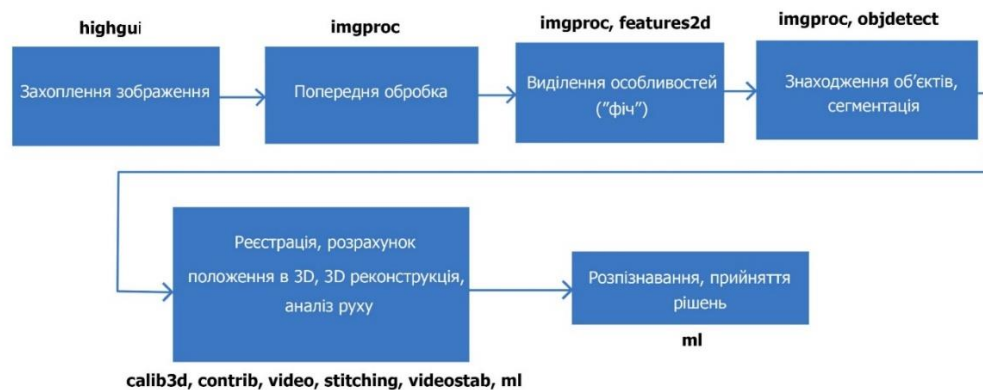


Рисунок 3.2 – Графічне представлення модулів бібліотеки openCV

Все починається з захвату зображення (модуль `highgui`). Зображення зчитується з файлу або відео з потокової камери через мережевий протокол.

Далі здійснюється попередня обробка (модуль `imgproc`), така, як усунення шуму, вирівнювання яскравості, контрасту, виділення і видалення

відблисків, тіней.

Наприклад, один і той же об'єкт при різному освітленні виглядає по-різному. У яскравому світлі червона машина, рух якої, необхідно відстежувати, буде яскраво-оранжевою. У похмуру погоду та ж машина буде виглядати червоно-рожевою. У цьому випадку на зображенні необхідно виконати вирівнювання кольору. Попередня обробка може бути простою, але може містити в собі складну технологію.

Наступний етап – виділення особливостей (модулі `imgproc`, `features2d`). Наприклад, в завданні стеження за об'єктом це може бути пошук спеціальних точок на об'єкті, за якими легко спостерігати; для завдання детектування (тобто виявлення на зображенні) обличчя - обчислення опису кожного пікселя.

Далі відбувається детектування цікавих для нас об'єктів, виділення значущих частин, сегментація зображення (модулі `imgproc`, `objdetect`). Якщо, наприклад, камера нерухома, а зображення рухоме, можна використовувати алгоритми віднімання фону.

Після цього вирішується основне завдання, таке, як обчислення розташування об'єкта в 3d, реконструкція 3d структури, аналіз структури, реєстрацію і т. П. (Модулі `calib3d`, `contrib`, `video`, `stitching`, `videostab`, `ml`).

Наприклад в задачі склейки панорам зображень - це зіставлення частин різних кадрів, визначення потрібного перетворення. У задачі відеоспостереження це відновлення траєкторій об'єктів і т. д. Наприкінці відбувається розпізнавання і прийняття конкретних рішень (модуль `ml`). Наприклад, в системі відеоспостереження: з'явився небажаний об'єкт в кадрі чи ні. У задачі детектування тексту - визначити текст, що саме за текст і т. д. [3].

3.3 Морфологічні зміни зображення

Для вирішення недоліку високого рівня шуму використовується морфологічні зміни зображення, які змінюють форму ключових об'єктів на

зображенні [20].

Алгоритм реалізації морфологічних операцій полягає в наступному. Створюється масив такого самого типу як зображення, який має такі самі розміри як вихідне зображення. У нього записується результат морфологічної операції. Обирається вікно, розміром з структурний елемент. Вікно переміщується по піксельно по всьому зображенні і для точки, яка опиняється в центрі вікна обирається мінімальне чи максимальне (в залежності від виконаної операції, ерозії чи нарощування) значення із значень пікселей, які знаходяться в межах вікна і це значення записується в масив.

Розглянемо такий спосіб як ерозія. Ерозія(звуження) зображення зазвичай використовується для позбавлення від випадкових вкраплень на зображенні. Ідея полягає в тому, що вкраплення при розмиванні усуваються, тоді як крупні і відповідно більш візуально-значущі регіони залишаються. Це операція зменшує границі форми. Операція для потовщення називається розширення. Ця операція збільшує границі для всіх форм на зображення, усуває шум на зображенні та сприяє об'єднанню областей, які були розділені шумом. Над наступним зашумленим зображенням виконаємо декілька морфологічних змін.

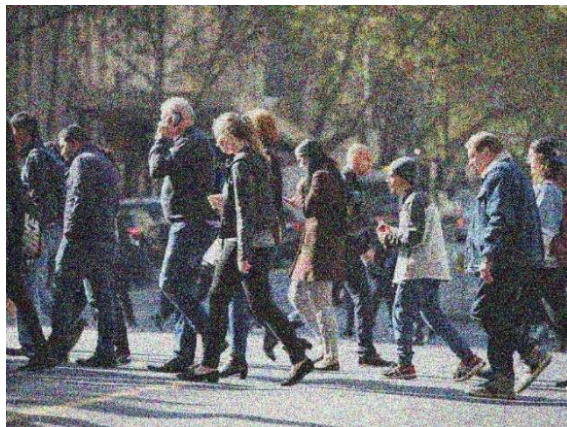


Рисунок 3.3 – Оригінальне зображення

Для подавлення шумів на зображенні розробимо та впровадимо

функцію `erosion()`, яка зображена на рис. 3.4.

```
erosion_size = 0
max_elem = 2
max_kernel_size = 21
title_trackbar_element_type = 'Element:\n 0: Rect \n 1: Cross \n 2: Ellipse'
title_trackbar_kernel_size = 'Kernel size:\n 2n +1'
title_erosion_window = 'Erosion Demo'
title_dilatation_window = 'Dilation Demo'

def erosion(val):
    erosion_size = cv2.getTrackbarPos(title_trackbar_kernel_size, title_erosion_window)
    erosion_type = 0
    val_type = cv2.getTrackbarPos(title_trackbar_element_type, title_erosion_window)
    if val_type == 0:
        erosion_type = cv2.MORPH_RECT
    elif val_type == 1:
        erosion_type = cv2.MORPH_CROSS
    elif val_type == 2:
        erosion_type = cv2.MORPH_ELLIPSE
    element = cv2.getStructuringElement(erosion_type, (2 * erosion_size + 1, 2 * erosion_size + 1),
                                       (erosion_size, erosion_size))
    erosion_dst = cv2.erode(dst, element)
    cv2.imshow(title_erosion_window, erosion_dst)
```

Рисунок 3.4 – Функція для виконання ерозії

Після впровадження функції `erosion` отримаємо зображення (рис. 3.5).



Рисунок 3.5 – Зображення після використання ерозії

Операція розмикання допомагає видалити з зображення елементи менші за структурний елемент, але це зумовлює нарощення контуру. Щоб уникнути цього застосовується операція замикання. Тому реалізуємо

функцію `dilatation` одразу після виконання ерозії (рис. 3.6).

```
def dilatation(val):
    dilatation_size = cv2.getTrackbarPos(title_trackbar_kernel_size, title_dilatation_window)
    dilatation_type = 0
    val_type = cv2.getTrackbarPos(title_trackbar_element_type, title_dilatation_window)
    if val_type == 0:
        dilatation_type = cv2.MORPH_RECT
    elif val_type == 1:
        dilatation_type = cv2.MORPH_CROSS
    elif val_type == 2:
        dilatation_type = cv2.MORPH_ELLIPSE
    element = cv2.getStructuringElement(dilatation_type, (2 * dilatation_size + 1, 2 * dilatation_size + 1),
                                       (dilatation_size, dilatation_size))
    dilatation_dst = cv2.dilate(dst, element)
    cv2.imshow(title_dilatation_window, dilatation_dst)
```

Рисунок 3.6 – Функція для виконання розширення

Після впровадження функції `dilatation`, отримаємо зображення (рис. 3.7).



Рисунок 3.7 – Зображення після використання розширення

3.4 Модифікація алгоритму розпізнавання об'єктів

Автоматичне виділення рухомих об'єктів сцени з вхідного відеопотоку – одна з найважливіших задач аналізу зображень. За останній час було запропоновано велику кількість методів для вирішення цього завдання за умови нерухомості відеокамери.

У представленій роботі запропоновано метод націлений на покращення процесу розпізнавання, в його основі лежить виділення об'єктів

на зображенні, ідентифікація кожного об'єкта відповідно до одного з 20 класів та визначення точності.

```
# Labels of network.
classNames = {0: 'background',
              1: 'aeroplane', 2: 'bicycle', 3: 'bird', 4: 'boat',
              5: 'bottle', 6: 'bus', 7: 'car', 8: 'cat', 9: 'chair',
              10: 'cow', 11: 'diningtable', 12: 'dog', 13: 'horse',
              14: 'motorbike', 15: 'person', 16: 'pottedplant',
              17: 'sheep', 18: 'sofa', 19: 'train', 20: 'tvmonitor'}
```

Рисунок 3.8 – Класи для ідентифікації об'єктів

Основні труднощі аналізу відеопотоку в реальному часі полягають в дуже великому обсязі даних при обмеженому часі для прийняття рішення. Щоб знайти в відеопотоці необхідний нам об'єкт, необхідно спочатку його виявити в режимі реального часу, а потім розпізнати. Ідентифікувати об'єкт будемо за допомогою алгоритма SSD(Single Shot MultiBox Detector).

Все починається з захвату зображення (модуль `highgui`). Зображення зчитується з файлу або відео з потокової камери через мережевий протокол. Далі здійснюється попередня обробка (модуль `imgproc`), така, як усунення шуму, вирівнювання яскравості, контрасту, виділення і видалення відблисків, тіней. Наприклад, один і той же об'єкт при різному освітленні виглядає по-різному.

Для фільтрації шуму будемо використовувати функцію `fastNLMMeansDenoisingColored()`.

```
# убираем шум и сохраняем изображение в dst
dst = cv2.fastNlMeansDenoisingColored(frame, None, 10, 10, 7, 21)
```

Рисунок 3.9 – Фрагмент коду для зменшення шуму зображення

Функція `fastNLMMeansDenoisingColored()` працює для кольорових зображень. Вона приймає такі аргументи:

- `h`: параметр, що визначає міцність фільтра. Більше значення `h` краще знімає шум, але також видаляє деталі зображення (рекомендовано 10);

- `hForColorComponents`: те саме, що `h`, але лише для кольорових зображень. (як правило, таке саме, як `h`);
- `templateWindowSize`: має бути непарним (рекомендується 7);
- `searchWindowSize`: має бути непарним (рекомендується 21).

В більшості програмних продуктах для обробки зображень використовують гама-корекцію. Якщо зображення необхідно освітлити чи затемнити, можна використовувати гама-корекцію. Наприклад, при освітленні можлива поява нових деталей в темних областях, які раніше не були помітні. Реалізуємо функцію для виконання гама-корекції для кожного кадру відеопослідовності:

```
def adjust_gamma(image, gamma=1.0):

    invGamma = 1.0 / gamma
    table = np.array([((i / 255.0) ** invGamma) * 255
        for i in np.arange(0, 256)]).astype("uint8")

    return cv2.LUT(image, table)
```

Рисунок 3.10 – Фрагмент коду для гама-корекції кадру

Для зниження шуму зображення і зменшення неточних визначень об'єктів, наступним кроком виконуються морфологічні операції. Послідовне використання операцій розмикання та замикання. Операція розмикання допомагає видалити з зображення елементи менші за структурний елемент, але це зумовлює нарощення контуру. Щоб уникнути цього застосовується операція замикання.

Вище зазначені методи й операції дозволяють безпосередньо перейти до визначення об'єктів.

Алгоритм розроблено за умови, що камера, яка передає відео послідовність нерухома. Алгоритм може оброблювати зображення з частотою кадрів камери.

Розмір об'єктів, що визначаються не менше 10 пікселів висотою і

шириною. Рух об'єкта поступальний по прямолінійній або криволінійній траєкторії, або періодичний. Швидкість руху не більше 100 пікселів на кадр.

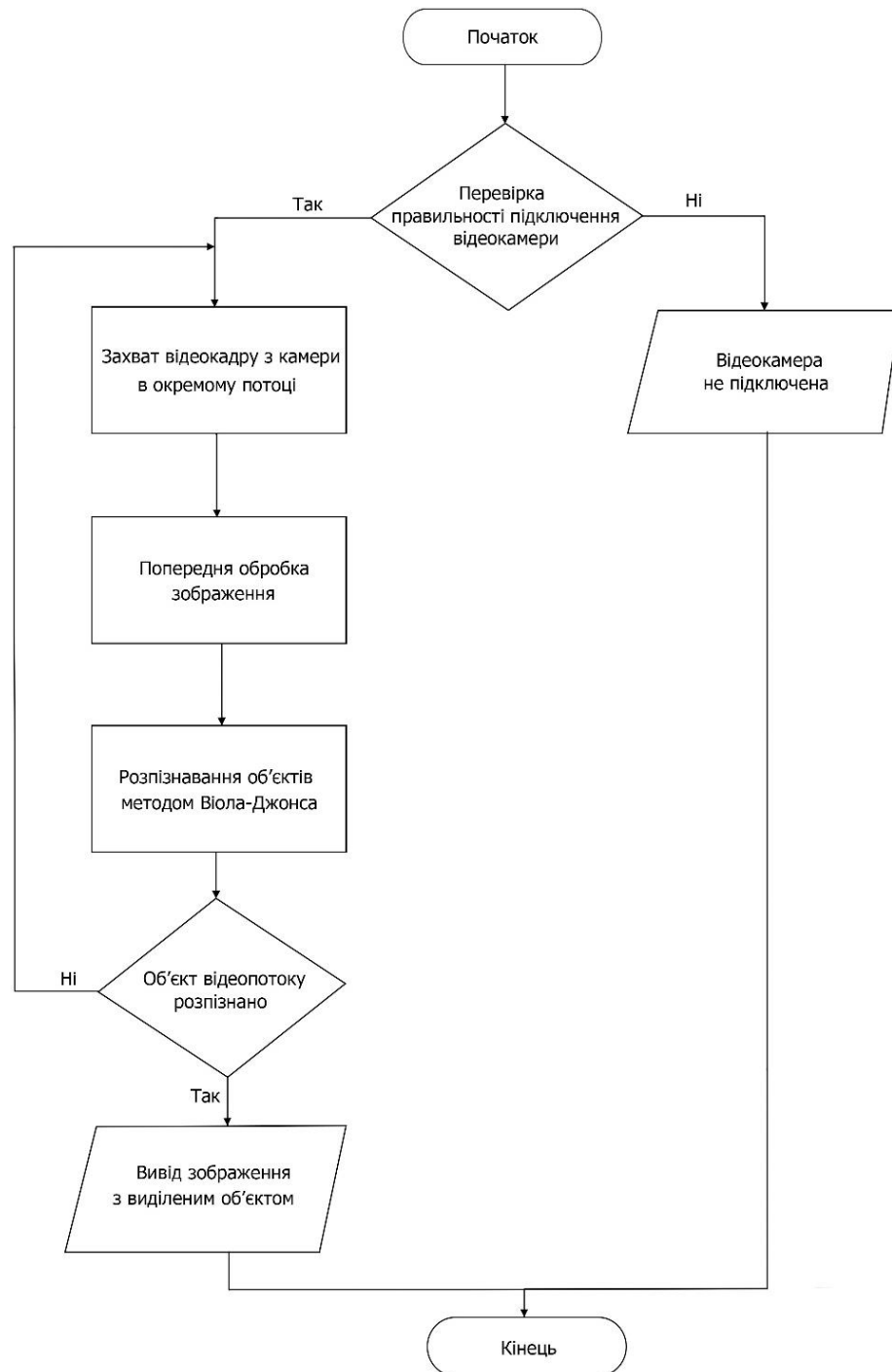


Рисунок 3.11 – Узагальнена схема роботи підсистеми

3.5 Підвищення якості відеотрекінгу за рахунок використання багатопоточності

Частота кадрів, або кадрова частота — це частота (швидкість), з якою пристрій формування зображення відображає послідовні зображення, що називаються кадрами.

Термін застосовується до кіно- і відео- камер, комп'ютерної графіки, і систем захоплення руху. Термін вперше використаний фотографом Едвардом Майбріджем, який здійснював експерименти з хронографічної зйомки рухомих об'єктів послідовно декількома фотокамерами. Частота кадрів зазвичай задається в кадрах за секунду (англ. frames per second, FPS).

Багатонитковість (англ. multi-threading), або багатопотоковість — властивість операційної системи або застосунку, яка полягає в тому, що процес, породжений в операційній системі, може складатися з кількох ниток, що виконуються паралельно, або навіть одночасно на багатопроцесорних системах. При виконанні деяких завдань таке розділення може досягти ефективнішого використання ресурсів комп'ютера. Такі нитки виконання ще називають потоками.

Суттю багатонитковості є квазі-багатозадачність на рівні одного виконуваного процесу, тобто всі нитки виконуються в адресному просторі процесу. Окрім цього, всі нитки процесу мають не тільки спільний адресний простір, але і спільні дескриптори файлів. Процес, що виконується, має як мінімум одну (головну) нитку.

Багатонитковість (як доктрину програмування) не слід плутати ані з багатозадачністю, ані з багато-процесорністю, не зважаючи на те, що операційні системи, що реалізують багатозадачність, як правило реалізують і багатонитковість.

Переваги в багатонитковості такі:

1. Спрощення програми в деяких випадках, за рахунок використання загального адресного простору;

2. Менші відносно процесу часові витрати на створення нитки і взаємодію між нитками;

3. Підвищення продуктивності процесу за рахунок розпаралелювання процесорних обчислень і операцій вводу/виводу.

Для покращення FPS створимо додатковий потік, який зчитуватиме з камери нові кадри, в той час як наш основний потік робитиме обробку поточного кадру.

Для отримання більш високої FPS при обробці відеопотоків за допомогою OpenCV будемо переміщувати ввід-вивід (тобто зчитування кадрів з датчика камери) в окремий потік. Процеси вводу / виводу зазвичай досить повільні. Незважаючи на те, що програми для комп'ютерного зору та обробки відео, як правило, досить важкі для процесора (особливо якщо вони призначені для роботи в режимі реального часу), введення / виведення кадрів камери також можуть бути важкими.

Для підвищення FPS перенесемо зчитування кадрів з веб-камери на зовсім інший потік, повністю відокремлений від нашого основного потоку. Це дозволить безперервно зчитувати кадри з потоку вводу/виводу, поки наш основний потік обробляє поточний кадр. Після того, як основний потік закінчить обробку кадру, він перехопить поточний кадр з потоку вводу/виводу. Це здійснюється без необхідності чекати блокування операцій вводу / виводу.

Перший крок у впровадженні нашої функціональності потокового відеопотоку - це визначення класу FPS, який ми можемо використовувати для вимірювання кількості кадрів у секунду. Цей клас допоможе нам отримати кількісні докази того, що потік дійсно збільшує FPS.


```

1  import datetime
2
3  class FPS:
4      def __init__(self):
5          # store the start time, end time, and total number of frames
6          # that were examined between the start and end intervals
7          self._start = None
8          self._end = None
9          self._numFrames = 0
10
11     def start(self):
12         # start the timer
13         self._start = datetime.datetime.now()
14         return self
15
16     def stop(self):
17         # stop the timer
18         self._end = datetime.datetime.now()
19
20     def update(self):
21         # increment the total number of frames examined during the
22         # start and end intervals
23         self._numFrames += 1
24
25     def elapsed(self):
26         # return the total number of seconds between the start and
27         # end interval
28         return (self._end - self._start).total_seconds()
29
30     def fps(self):
31         # compute the (approximate) frames per second
32         return self._numFrames / self.elapsed()

```

Рисунок 3.12 – Фрагмент коду для класа FPS

Потім ми визначимо клас WebcamVideoStream, який матиме доступ до нашої веб-камери. Визначимо наш сценарій, fps_demo.py, який буде порівнювати однопотоковий та багатопотоковий FPS.

Таблиця 3.1 Порівняння швидкодії підсистеми при роботі в одному потоці та винесення зчитування даних в окремий потік

№ Випробування	Швидкість обчислення, fps	
	Зчитування і обробка кадрів в одному потоці	Виокремлення зчитування даних в окремий потік
1	23,65	70,95
2	23,66	71,64
3	23,67	70,84

```

1  # import the necessary packages
2  from threading import Thread
3  import cv2
4
5  class WebcamVideoStream:
6      def __init__(self, src=0):
7          # initialize the video camera stream and read the first frame
8          # from the stream
9          self.stream = cv2.VideoCapture(src)
10         (self.grabbed, self.frame) = self.stream.read()
11
12         # initialize the variable used to indicate if the thread should
13         # be stopped
14         self.stopped = False
15
16     def start(self):
17         # start the thread to read frames from the video stream
18         Thread(target=self.update, args=()).start()
19         return self
20
21     def update(self):
22         # keep looping infinitely until the thread is stopped
23         while True:
24             # if the thread indicator variable is set, stop the thread
25             if self.stopped:
26                 return
27
28             # otherwise, read the next frame from the stream
29             (self.grabbed, self.frame) = self.stream.read()
30
31     def read(self):
32         # return the frame most recently read
33         return self.frame
34
35     def stop(self):
36         # indicate that the thread should be stopped
37         self.stopped = True
38

```

Рисунок 3.13 – Фрагмент коду для класа WebcamVideoStream

Як ми бачимо, послідовно зчитуючи кадри з нашого відеопотоку в основному потоці нашого сценарію, ми можемо отримати 23,65 FPS. Однак, як тільки ми переходимо до використання окремого потоку для вводу кадрів, ми досягаємо 71,64 FPS - збільшення у 3 рази.

3.6 Тестування розробленої підсистеми

Для оцінки результатів роботи будемо розглядати наскільки точно програма розпізнає об'єкти, що рухаються. Тобто кількість правильно

визначених об'єктів і шуму. Програма працює тільки з послідовностями зображень, які мають статичний фон. Деякі зміни у фоні, наприклад, коливання дерев чи снігопад зумовлюють їх розпізнавання як об'єктів переднього фону. Неякісне зображення може привести до помилкових визначень об'єктів, що також дає негативний результат. Отже, емпіричним шляхом проаналізуємо результати роботи програми.

Приклад № 1

Розглянемо приклад розпізнавання чотирьох об'єктів на відеопотоці в реальному часі.

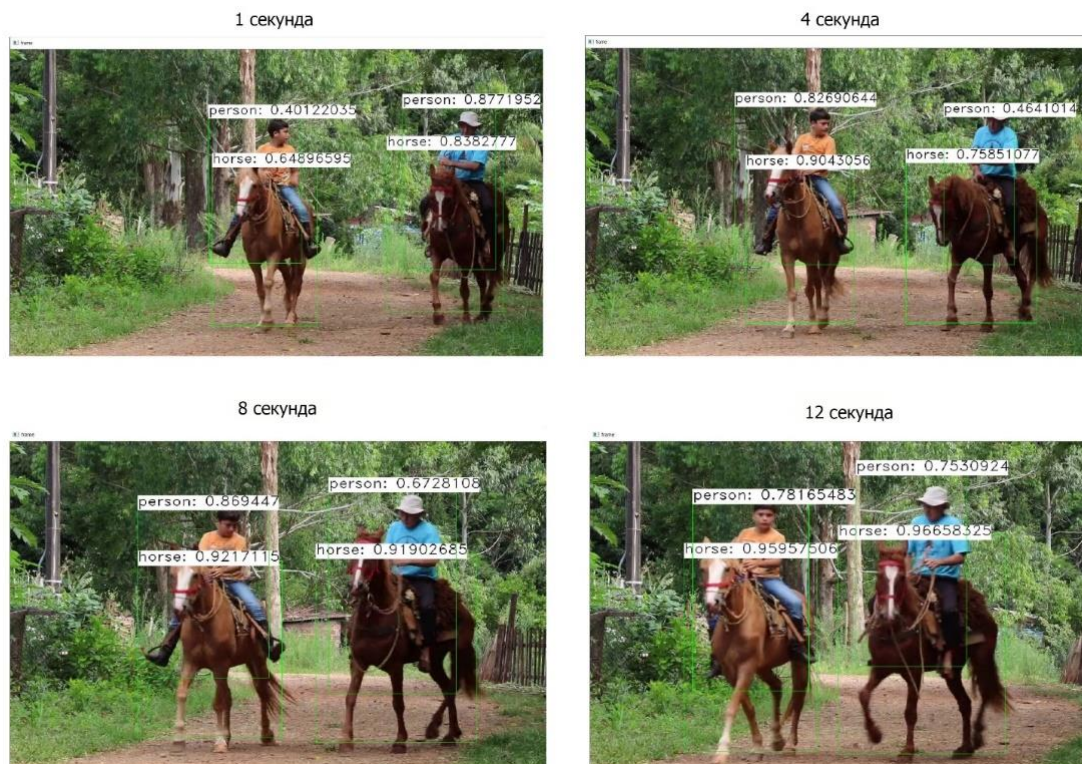


Рисунок 3.14 – Зображення роботи програми

Програма виводить відеопослідовність з розпізнаним об'єктом. Визначені об'єкти позначаються прямокутниками зеленого кольору, мають назву та точність розпізнавання.

Дана послідовність містить чотири об'єкти, що рухаються по криволінійній траєкторії. Протягом усієї послідовності програма надає можливість бачити зміни положення об'єктів, що відслідковуються. На початку відстеження програма правильно розпізнала чотири об'єкти з

точностями 0,4012%, 0,6489%, 0,8771% та 0,8382%. Через 12 секунд точність ідентифікації розпізнаних об'єктів значно зростає. Таблиця порівняння точностей розпізнавання об'єктів кожні 4 секунди наведено в табл. 3.2.

Таблиця 3.2 – Точність розпізнавання об'єктів через рівні проміжки часу

Час, сек	Точність розпізнавання, %			
	Об'єкт 1	Об'єкт 2	Об'єкт 3	Об'єкт 4
1	40,12	64,89	87,71	83,82
4	82,69	90,43	46,41	75,85
8	86,94	92,17	67,28	91,90
12	78,16	95,95	75,30	96,65

Результати програми повністю співпадають з вимогами: знайдені об'єкти правильно ідентифіковані. З приближенням об'єктів точність їх розпізнавання підвищується. Отже, незважаючи на кількість об'єктів, шум реальної камери та часткове перекриття об'єктів, результат даного прикладу повністю задовольняє очікування.

Приклад № 2

Розмір кадру даної послідовності зображення – 640 пікселів на 360. Розглядається лінійний рух декількох об'єктів. Ця послідовність містить сильний шум, що значно знизив точність розпізнавання в порівнянні з попереднім прикладами. На початку відстеження програма правильно визначила лише два об'єкта через перекриття (оклюзію) більше половини інших об'єктів. Наприкінці відстеження (12 секунда) програма правильно визначила три об'єкти, навіть при сильному зашумленні, рис. 3.15.



Рисунок 3.15 – Результат роботи програми

Таблиця порівняння точностей розпізнавання об'єктів кожні 4 секунди наведено в табл. 3.3.

Таблиця 3.3 – Точність розпізнавання об'єктів через рівні проміжки часу

Час, сек	Точність розпізнавання, %			
	Об'єкт 1	Об'єкт 2	Об'єкт 3	Об'єкт 4
1	74,34	97,82	Не розпізнано	Не розпізнано
4	90,02	86,09	30,92	Не розпізнано
8	58,96	94,01	Не розпізнано	52,24
12	58,96	94,01	Не розпізнано	52,24

Результати програми повністю співпадають з вимогами: знайдені об'єкти правильно ідентифіковані. З приближенням об'єктів точність їх розпізнавання підвищується. Об'єкт №3 на 1, 8 і 12 секундах не ідентифікований через перекриття об'єктом №1.

Об'єкт №4 не був розпізнаний на 1 і 4 секундах через сильне зашумлення кадрів. Отже, незважаючи на кількість об'єктів, шум реальної

камери та часткове перекриття об'єктів, програма правильно визначила 4 об'єкти.

Приклад № 3

Розмір кадру даної послідовності зображення – 640 пікселів на 360. Розглядається повільний, поступальний рух декількох об'єктів. Ця послідовність містить зміну інтенсивності освітлення, що знизило точність розпізнавання порівняно з нормальними умовами. Проаналізуємо якість розпізнавання і визначення об'єктів на наступному прикладі.



Рисунок 3.16 – Результат роботи програми

Дана послідовність містить два об'єкти, що рухаються по прямолінійній траєкторії. Протягом усієї послідовності програма надає можливість бачити зміни положення об'єктів, що відслідковуються. На початку відстеження програма правильно розпізнала один об'єкт з точністю 0,6533%, другий об'єкт знаходився в кадрі наполовину.

Через 12 секунд точність ідентифікації розпізнаних об'єктів значно зросла. Таблиця порівняння точностей розпізнавання об'єктів кожні 4 секунди наведено в табл. 3.4.

Таблиця 3.4 – Точність розпізнавання об'єктів через рівні проміжки часу

Час, сек	Точність розпізнавання, %	
	Об'єкт 1	Об'єкт 2
1	65,33	Не розпізнаний
4	Не розпізнаний	81,95
8	Не розпізнаний	97,93
12	Не розпізнаний	99,51

В результаті роботи програми були розпізнані два об'єкти в різні проміжки часу. На 4 секунді об'єкт №1 не був ідентифікований через значну зміну інтенсивності освітлення. З приближенням об'єктів точність її розпізнавання підвищувалася.

Порівняємо результати роботи існуючого методу розпізнавання та розробленого модифікованого методу розпізнавання об'єктів. Для цього розрахуємо середнє значення відсотку розпізнаних об'єктів модифікованим методом та існуючим при чотирьох умовах: ідеальних умовах розпізнавання, при частковому перекритті, зашумлених відеопослідовності та при значній зміні інтенсивності освітлення.

Можна зробити висновок, що шляхом комбінування методу медіанної фільтрації, морфологічних операцій, із застосуванням сегментації на основі порогової обробки відеопотоку відсоток розпізнавання об'єктів став дещо вищим. Отже, якість розпізнавання об'єктів на відеопослідовності в режимі реального часу стала вищою.

Таблиця 3.5 Порівняння існуючого алгоритму розпізнавання та алгоритму з попередньою обробкою кадрів

Умови тестування	Точність розпізнавання, %		
	Модифікований метод розпізнавання об'єктів	Вихідний метод розпізнавання об'єктів	Відсоток покращення
Ідеальні умови розпізнавання	92,14	92,14	Не змінився
Сильний шум	82,63	90,00	7.37
Зміна інтенсивності освітлення	55,12	63,45	8.33

Висновки за розділом

В подальшому для покращення результату планується впровадити можливість розпізнавання та трекінгу об'єктів в реальному часі з використанням рухомої камери та навчити нейронну сітку розпізнавати більшу кількість класів. Основним недоліком алгоритму є розпізнавання шуму як об'єкта переднього плану. Для того, щоб уникнути таких помилкових визначень можна використовувати різні способи фільтрації об'єктів:

- для пошуку об'єктів використовувати співставлення за шаблоном. Кожне зображення порівнюється з зображенням шаблона для локалізації об'єкта. Такими шаблонами можуть бути силуети людини, машини тощо;
- зробити навчання каскада Хаара для підвищення точності розпізнавання при значному перекритті об'єктами;
- застосування багатокамерного трекінгу для співставлення об'єктів на різних кадрах одного моменту часу. Це потребує точної калібровки

камери. Дозволяє позбавитись недоліків оклюзії одного об'єкта іншим;

Майже кожний з розглянутих вище методів потребує збільшення навантаження на процесор для підрахування додаткових параметрів. Це суттєво знижує ефективність трекінгу об'єктів в реальному часі, але підвищує якість визначення і відстеження.

4. РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

4.1 Опис ідеї проекту

Завдання розробки алгоритмів систем відеоспостереження в даний час є надзвичайно актуальною і зачіпає багато сторін життя людини. З появою дешевих цифрових відеореєстраторів стало можливим обробляти вимірювальні дані за допомогою комп'ютера. Прикладами таких завдань є охорона периметра і внутрішньої території об'єкта, виявлення і розпізнавання осіб, розпізнавання руху. Системи відеоспостереження знаходять широке застосування в задачах боротьби з тероризмом і моніторингу дорожнього руху. Великий інтерес в області алгоритмів супроводження руху зумовило поширення потужних комп'ютерів, доступність високоякісних камер за невеликою ціною та збільшення потреб для автоматизованого аналізу відео. Основні ключові кроки відеоаналітики: розпізнавання об'єкта, що рухається, відстеження об'єкта кадр за кадром, і аналіз об'єктів для визначення їх поведінки.

Опираючись на те, що було сказано вище, ідея стартап-проекту полягає у створенні підсистеми для підвищення якості процесу розпізнавання об'єктів відеопотоку шляхом використання комбінованого метода ідентифікації.

Для виконання даного підпункту було розглянуто та проаналізовано, а також зведено у таблицях:

1. зміст ідеї;
2. можливі напрямки застосування;
3. основні вигоди, що може отримати користувач товару (за кожним напрямом застосування);
4. основні відмінності від вже існуючих рішень.

Таблиця 4.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розроблену підсистему можна застосувати до багатьох систем відеоаналітики, а також в роботизованих системах	1. Автоматизоване спостереження, що виявляє підозрілу активність	Можливість використання програмного забезпечення з отриманням високої точності розпізнавання, навіть при неякісних вхідних даних в режимі реального часу
	2. Взаємодія людина-комп'ютер, наприклад, слідкування за поглядом для вводу даних, розпізнавання жестів	
	3. Навігація машин, що пов'язана з планування маршруту та униканням перешкод.	

Аналіз потенційних техніко-економічних переваг ідеї (чим відрізняється від існуючих аналогів та замінників) порівняно із пропозиціями конкурентів передбачає:

- визначення переліку техніко-економічних властивостей та характеристик ідеї;
- визначення попереднього кола конкурентів (проектів-конкурентів) або товарів-замінників чи товарів-аналогів, що вже існують на ринку, та проводиться збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку;
- проводиться порівняльний аналіз показників: для власної ідеї визначаються показники, що мають гірші значення (W, слабкі), аналогічні (N, нейтральні) значення, кращі значення (S, сильні) (табл. 4.2).

Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності.

Таблиця 4.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№	Техніко-економічні характеристики ідеї	Товари/концепції конкурентів		W (слабка сторона)	N (нейтральна сторона)	S(сильна сторона)
		Мій проект	Програмне забезпечення для розпізнавання			
1	Зручний інтерфейс	Програмний додаток має головне вікно, в якому відображається відеопоток з розпізнаними об'єктами, можливість зберегти зображення кадру.	Має головне вікно, в якому відображається відеопоток з розпізнаними об'єктами.			+
2	Ціна	Має низьку вартість	Тільки платне програмне забезпечення			+
3	Кросплатформеність	Працює на ОС Windows	Працює на ОС Windows		+	
4	Вихідний код	Невеликий розмір вихідного коду	Суттєвий розмір вихідного коду			+
5	Наявність у відкритому доступі	Наявне	Відсутнє		+	

4.2 Технологічний аудит ідеї проекту

В межах даного підрозділу було проведено аудит технології, за допомогою якої можна реалізувати ідею проекту (технології створення товару). Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (таблиця 4.3):

- а) за якою технологією буде виготовлено товар згідно ідеї проекту;
- б) чи існують такі технології, чи їх потрібно розробити/додати;
- в) чи доступні такі технології авторам проекту.

За результатами аналізу таблиці зроблено висновок щодо можливості технологічної реалізації проекту. Технологічним шляхом реалізації проекту було обрано такі технології, як бібліотека OpenCV мова програмування Python через їх доступність та безкоштовність.

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
Розроблення підсистеми для розпізнавання об'єктів відеопотоку	Середовище розробки PyCharm	Наявна	Доступна
	Бібліотека OpenCV	Наявна	Доступна
	Мова програмування Python	Наявна	Доступна

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів. Спочатку було проведено аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (табл. 4.4).

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап-проекту

Показники стану ринку (найменування)	Характеристика
Кількість головних гравців, од	2
Загальний обсяг продаж, грн/ум. Од	25000
Динаміка ринку (якісна оцінка)	Зростає
Наявність обмежень для входу (вказати характер обмежень)	-
Специфічні вимоги до стандартизації та сертифікації	-
Середня норма рентабельності у галузі (або по ринку), %	15

Середню норму рентабельності в галузі було порівняно із банківським відсотком на вкладення. Останній є меншим, тому є сенс вкладати гроші саме у цей проект. За результатами аналізу табл. 4.4 було зроблено

висновок, що ринок є привабливим для входження. Надалі були визначені потенційні групи клієнтів, їх характеристики та зформовано орієнтовний перелік вимог до товару для кожної групи (табл. 4.5).

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
Підсистема для розпізнавання об'єктів відеопотоку в реальному часі	Компанії, приватні підприємства специфіка роботи яких пов'язана з використанням комп'ютерного зору	Відмінності у сферах діяльності компаній та приватних підприємств	Прийнятна цінова політика, зручний інтерфейс.

Після визначення потенційних груп клієнтів було проведено аналіз ринкового середовища: складено таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл. 4.6, 4.7).

Таблиця 4.6 – Фактори загроз

Фактор	Зміст загрози	Можлива реакції компанії
Конкуренція	Вихід на ринок продуктів з кращими показниками	Вдосконалення використовуваних алгоритмів Популяризація продукту за рахунок рекламної кампанії
Зростання вимог користувачів	Користувачам необхідна підсистема, що буде забезпечувати захист від нових видів загроз	Розширення функціоналу продукту Популяризація продукту за рахунок рекламної кампанії

Визначено можливу реакцію компанії на конкуренцію, зміну потреб користувачів та появу нових методів з врахуванням можливостей ринку та вітчизняних особливостей.

Таблиця 4.7 – Фактори можливостей

Фактор	Зміст можливості	Можлива реакції компанії
Конкуренція	Майже повна відсутність аналогів	
Поява нових способів	Нові методи для підвищення якості розпізнавання	Оптимізація існуючого рішення впровадженням нових алгоритмів та способів

Надалі було проведено аналіз пропозиції: визначили загальні риси конкуренції на ринку (табл. 4.8). Було проведено аналіз конкуренції у галузі за моделлю М. Портера (табл. 4.9).

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємництва (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції – монополія	На ринку присутні декілька компаній-конкурентів, але їх товар дещо відрізняється між собою.	Підтримка якості продукту та постійні вдосконалення
2. За рівнем конкурентної боротьби – міжнародний	Компанії-конкуренти з інших країн	Розробити універсальну архітектуру продукту, що дозволить легко адаптувати продукт до інших галузей
3. За галузевою ознакою – міжгалузева	Продукт може бути використаний для інших галузей	Постійне вдосконалення та оновлення продукту, що сфокусовано на первісну ціль
4. Конкуренція за видами товарів – товарно-видова	Конкуренція між видами програмного продукту, їх якістю та оновленням	Створити програмний продукт, враховуючи недоліки конкурентів та актуальність функціоналу
5. За характером конкурентних переваг – нецінова	Вдосконалення технології створення продукту для низької собівартості	Покращення моделі функціонування продукту. Використання дешевих актуальних технологій, що дозволяють дотриматись вимог якості продукту
6. За інтенсивністю – не марочна	Бренд присутній, але його роль незначна	Проведення рекламних кампаній

За результатами аналізу табл. 4.9 було зроблено висновок про можливість роботи на ринку з огляду на конкурентну ситуацію. Також було

зроблено висновок щодо характеристик, які повинен мати проект, щоб бути конкурентоспроможним на ринку.

Цей висновок був врахований при формулюванні переліку факторів конкурентоспроможності у наступному пункті. На основі аналізу конкуренції, проведеного в табл. 4.9, а також із урахуванням характеристик ідеї проекту (табл. 4.2), вимог споживачів до товару (табл. 4.5) та факторів маркетингового середовища (табл. 4.6, 4.7) визначається та обґрунтовується перелік факторів конкурентоспроможності.

Таблиця 4.9 – Аналіз конкуренції у галузі за М. Портером

Складові аналізу	Прямі конкуренти у галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Навести перелік прямих конкурентів	Визначити бар'єри входження в ринок	Визначити фактори сили постачальників	Визначити фактори сили споживачів	Фактори загроз з боку замінників
	Програмне забезпечення для розпізнавання	Наявність вже існуючих рішень	-	Контроль якості продукту	Авторитет конкурентів на ринку, більш широкий функціонал
Висновки	Досить інтенсивна боротьба з вже закріпившимися на ринку гравцями	Є можливість виходу на ринок, але є конкуренти. Строки – 12 місяців	-	Споживачі диктують умови на ринку: надійний, швидкий та точний програмний продукт для захисту веб-орієнтованих систем	Необхідно випускати програмний продукт не гірше, ніж у конкурентів та розширювати функціонал

За визначеними факторами конкурентоспроможності (табл. 4.10) проведено аналіз сильних та слабких сторін стартап-проекту (табл. 4.11).

Аналіз оформлено у табл. 4.10.

Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для конкурентних проектів значущим)
Цінова політика	Оптимальна – товар доступний для значної кількості споживачів.
Висока якість	Програмне забезпечення розроблено згідно стандарту і є ліцензованим.
Актуальність	Реалізована підсистема

	актуальна у використанні в різних галузях
--	---

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін проекту

Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні						
		-3	-2	-1	0	1	2	3
Цінова політика	15				+			
Висока якість	20						+	
Актуальність	15		+					

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (табл. 4.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл. 4.11). Перелік ринкових загроз та ринкових можливостей було складено на основі аналізу факторів загроз та факторів можливостей маркетингового середовища.

Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення. Наприклад: зниження доходів потенційних споживачів – фактор загрози, на основі якого можна зробити прогноз щодо посилення значущості цінового фактору при виборі товару та відповідно, – цінової конкуренції (а це вже – ринкова загроза).

На основі SWOT-аналізу було розроблено альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок (див. табл. 4.9, аналіз потенційних конкурентів). Визначені альтернативи були проаналізовані з точки зору строків та ймовірності отримання ресурсів (табл. 4.13).

Таблиця 4.12 – SWOT-аналіз стартап-проекту

Сильні сторони:	Слабкі сторони:
Поширення реклами, яка буде ознайомлювати користувачів з новим товаром Застосування новітніх технологій при розробці Залучення професійних і кваліфікованих кадрів	Недостатній імідж, оскільки програмний продукт тільки починає роботу на ринку
Можливості:	Загрози:
Збільшення попиту на продукцію Залучення висококваліфікованого персоналу Більшість клієнтів зацікавлені у використанні програмного продукту	Нестабільна економічна і політична ситуація у країні Наявність прямих конкурентів

Таблиця 4.13 – Альтернативи ринкового впровадження стартап-проекту

Альтернатива (орієнтований комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
Безкоштовне розповсюдження створеного продукту	70%	12 місяців
Створення продукту з подальшим розповсюдженням за певну оплату	75%	12 місяців

Після аналізу було обрано альтернативу No2.

4.4 Аналіз ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: було проведено опис цільових груп потенційних споживачів (табл. 4.14).

За результатами аналізу потенційних груп споживачів було обрано цільові групи, для яких буде запропоновано даний товар, та визначено стратегію охоплення ринку - стратегію диференційованого маркетингу(компанія працює з декількома сегментами). Як цільові групи обрано: 1,2,3.

Таблиця 4.14 – Вибір цільових груп потенційних споживачів

Опис профілю цільової групи потенційних клієнтів	Готовність поживачі в прийняти продукт	Орієнтований попит в межах цільової групи (сегменти)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
Компанії діяльність яких пов'язана з системами розпізнавання	Висока	Високий	Сильна	Складно
Приватні підприємства міського та міжнародного рівня, діяльність яких пов'язана з системами розпізнавання	Висока	Високий	Сильна	Складно
Приватні підприємства, обласного рівня	Висока	Високий	Помірна	Середня складність
Підприємства регіонального характеру	Висока	Середній	Слабка	Просто

Для роботи в обраних сегментах ринку сформовано базову стратегію розвитку (табл. 4.15).

Таблиця 4.15 – Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Створення продукту з подальшим розповсюдженням за певну оплату	Визначити потреби кожної з груп, розробити відповідно до них стратегії приваблення клієнтів та маркетингової комунікації	Цінова політика, універсальність продукту (миттєве практичне застосування), якість та актуальність	Стратегія диференціації

Наступним кроком обрано стратегію конкурентної поведінки (табл. 4.16).

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
«першопроходець»	Забирати існуючих	Ні	Стратегія зайняття конкурентної ніші

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту (див. табл. 4.5), а також в залежності від обраної базової стратегії розвитку (табл. 4.15) та стратегії конкурентної поведінки (табл. 4.16) розроблено стратегію позиціонування (табл. 4.17), що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 4.17 – Визначення стратегії позиціонування

Вимоги до товару цільовою аудиторією	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
Просте використання, надійність, швидкість, наявність оновлень та наявність документації для програмного продукту	Стратегія диференціації	Позиція на основі порівняння фірми з товарами конкурентів; Відмінні особливості споживача	Надійність Швидкість Просте використання

Результатом виконання підрозділу стала узгоджена система рішень щодо ринкової поведінки стартап-компанії, яка визначає напрями роботи стартап-компанії на ринку.

4.5 Розроблення маркетингової програми стартап-проекту

Сформовано маркетингову концепцію товару, який отримає споживач. Для цього у табл. 4.18 підсумовано результати попереднього аналізу конкурентоспроможності товару. Концепція товару – письмовий опис фізичних та інших характеристик товару, які сприймаються споживачем, і набору вигод, які він обіцяє певній групі споживачів.

Таблиця 4.18 – Визначення ключових переваг концепції потенційного товару

Потреба	Вгода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
Швидкість обробки даних	Швидка обробка вхідних даних	Алгоритми реалізовані найоптимальнішим способом та забезпечують найменшу затримку при обробці
Зручність застосування	Не потребує складного налаштування для впровадження у систему	Заощаджує час впровадження у систему за рахунок створеної архітектури
Точність обробки даних	Висока точність ідентифікації об'єктів	Висока вірогідність виявлення об'єктів за рахунок використання сучасних алгоритмів

Розроблено трирівневу маркетингову модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (табл. 4.19). 1-й рівень: при формуванні задуму товару вирішується питання щодо того, засобом вирішення якої потреби або проблеми буде даний товар, яка його основна вигода. Дане питання безпосередньо пов'язаний з формуванням технічного завдання в процесі розробки конструкторської документації на виріб. 2-й рівень: цей рівень являє рішення того, як будереалізований товар в реальному, включає в себе якість, властивості, дизайн, упаковку, ціну. 3-й рівень: товар з підкріпленням (супроводом) – додаткові послуги та переваги для

споживача, що створюються на основі товару за задумом і товару в реальному виконанні (гарантії якості, доставка, умови оплати та ін).

Після формування маркетингової моделі товару слід відмітити, що проект буде захищено від копіювання за допомогою ноу-хау. Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар, яке передбачає аналіз ціни на товари-аналоги або товари субституту, а також аналіз рівня доходів цільової групи споживачів (табл. 4.20). Аналіз проведено експертним методом.

Таблиця 4.19 – Опис трьох рівнів моделі товару

Рівні товарів	Сутність та складові		
Товар за задумом	Швидке та точне розпізнавання об'єктів відеопотоку в реальному часі		
Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
Товар із підкріпленням	1.функція для предобробки вхідних даних		
	2. функція для зчитування кадрів в окремому потоці		
	Якість: точна ідентифікація об'єктів в реальному часі		
	Пакування: відсутнє		
Товар із підкріпленням	Марка: Detection		
	До продажу: відсутнє		
Товар із підкріпленням	Після продажу: персональна підтримка з можливістю розширення функціоналу під власні потреби		
	Вихідний код закритий. На ідею зареєстровано патент.		

Таблиця 4.20 – Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
50000 грн	44000 грн	У всіх трьох груп достатній рівень доходів	Базова покупка 30000 грн Подальша персональна підтримка в обслуговуванні 3000 грн

Наступним кроком є визначення оптимальної системи збуту, в межах якого було прийняте рішення (табл. 4.21).

Таблиця 4.21 – Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Цільові клієнти – компанії, які бажають впровадити у своїй роботі сучасні засоби, які забезпечать точність та швидкість розпізнавання об'єктів. Вони цікавляться існуючими рішеннями та інноваціями у сфері комп'ютерного зору.	Встановлення контактів із споживачами і підтримання їх. Формування попиту і стимулювання збуту. Дослідницька робота зі збору маркетингової інформації. Забезпечення зворотного зв'язку зі споживачами.	Один (від виробника одразу споживачу)	Прямий канал збуту до споживача, мінімізувати збутові витрати розвитку маркетингового спілкування із споживачем

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 4.22).

Результатом підрозділу стала ринкова (маркетингова) програма, що включає в себе концепції товару, збуту, просування та попередній аналіз можливостей ціноутворення, спирається на цінності та потреби потенційних клієнтів, конкурентні переваги ідеї, стан та динаміку ринкового середовища, в межах якого впроваджено проект, та відповідну обрану альтернативу ринкової поведінки.

Таблиця 4.22 – Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Цільові клієнти – компанії, які бажають впровадити у своїй роботі сучасні засоби, які забезпечать точність і швидкість ідентифікації об'єктів. Вони цікавляться існуючими рішеннями та інноваціями у сфері комп'ютерного зору	Конференції, інтернет-конференції, семінари, огляд професійної літератури, інтернет, періодичні видання у різноманітних (профільних) галузях.	Позиція на основі порівняння фірми з товарами конкурентів; Відмінні особливості споживача	Створення репутації фірмі —виробнику чи посереднику; збільшення чистого прибутку та рентабельності фірми; збільшення потоків покупців та обсягів продажу; стабілізація обсягів продажу в період зменшення попиту та загального спаду ділової активності.	

Висновки за розділом

В даному розділі було проведено аналіз програмного продукту у якості стартап проекту. Можна зазначити, що у проекту є можливість комерціалізації, оскільки ринок потребує якісний продукт, що надає можливість якісно і швидко розпізнавати об'єкти відеопотоку. На ринку наявна монополістична конкуренція, існує декілька фірм-конкурентів, але їх товар дещо відрізняється, тому вихід на ринок не буде легким і потребує грамотної стратегії виходу. Для впровадження ринкової реалізації проекту слід обрати альтернативу, яка передбачає розробку програмного продукту з подальшим розповсюдженням за певну плату. Можна сказати, що подальший розвиток проекту є доцільним, оскільки він знайде свою цільову аудиторію.

ВИСНОВКИ

У даній роботі здійснено аналіз існуючих методів розпізнавання та трекінгу об'єктів, а також розглянуті можливі способи покращення якості розпізнавання. Дослідження в даній магістерській роботі дали змогу зробити наступні висновки:

1. був проведений аналіз (ефективності та точності) існуючих методів виділення і розпізнавання об'єктів;
2. удосконалено алгоритм пошуку об'єктів відеопотоку, за рахунок комбінованого використання фільтрації шуму, гама-корекції, морфологічних операцій та використання багатопоточності для розпізнавання;
3. було розроблено програмне забезпечення розпізнавання об'єкта у відеопотоці на мові програмування Python;
4. проведено випробування розробленого програмного забезпечення для різних варіантів неякісних вхідних даних.

Таким чином за результатами магістерської роботи можна зробити висновок, що модифікований метод розпізнавання об'єктів дає можливість краще знаходити потрібні образи на зображеннях поганої якості та деталізації.

ПЕРЕЛІК ПОСИЛАНЬ

1. ИНТУИТ. URL: <http://www.intuit.ru/studies/courses/10621/1105/lecture/17985> (дата звернення 12.10.2019).
2. Розпізнавання об'єктів в реальному часі. URL: <https://proglib.io/p/real-time-object-detection/> (дата звернення 20.10.2019).
3. OpenCV library. URL: <https://opencv.org/> (дата звернення 15.09.2019).
4. NumPy. URL: <http://www.numpy.org/> (дата звернення 07.10.2019).
5. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs / [L.-C. Chen, G. Papandreou, I. Kokkinos, et al.] URL: <https://arxiv.org/pdf/1606.00915.pdf> (дата звернення 12.10.2019).
6. Вьюгин В.В. Математические основы теории машинного обучения и прогнозирования / В.В. Вьюгин. – М.: МЦНМО, 2013. 387 с.
7. Лукьяница А. А. Обработка видеоизображений / А.А.Лукьяница, А.Г.Шишкин; Под ред. Поздняков С.А. — М.: Ай-Эс-Эс Пресс, 2009. 764 с.
8. Алпатов Б.А., Бабаян П.В., Балашов О.Е., Степашкин А.И. Методы автоматического обнаружения и сопровождения объектов. Обработка изображений и управление. М., Радиотехника, 2008. 176 с.
9. Гансалес Р., Вудс Р. Цифровая обработка изображений. М., Техносфера, 2005. 1012 с.
10. Сойфер В. А. Методы компьютерной обработки изображений. М., ФИЗМАТЛИТ, 2003. 784 с.
11. Shi J., Tomasi C. Good features to track. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Seattle, USA, 1994, pp. 593–600.
12. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs / [L.-C. Chen, G. Papandreou, I. Kokkinos, et al.]. URL: <https://arxiv.org/pdf/1606.00915.pdf> (дата звернення 10.09.2019).
13. Comaniciu D., Ramesh V., Meer P. Kernel-based object tracking. IEEE

- Transaction on Pattern Analysis and Machine Intelligence, 2003, pp. 564–577.
14. Isard M., Blake A. Condensation — conditional density propagation for visual tracking. *International Journal of Computer Vision*, 1998, pp. 5–28.
 15. Freeman W.T., Roth M. Orientation histograms for hand gesture recognition. *Proceedings of the Workshop on Automatic Face and Gesture Recognition*. Zurich, Switzerland, 1995, pp. 296–301.
 16. Kwon J., Lee K.M. Visual Tracking Decomposition. *Conference on Computer Vision and Pattern Recognition*, 2010.
 17. Babenko B., Yang M.-H., Belongie S. Visual Tracking with Online Multiple Instance Learning. *Conference on Computer Vision and Pattern Recognition*, 2009.
 18. Lowe G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004, vol. 60, no. 2, pp. 91–110.
 19. Viola P., Jones M. Rapid object detection using a boosted cascade of simple features. *Conference on Computer Vision and Pattern Recognition*, 2001.
 20. Апальков И.В., Хрящев В.В. Удаление шума из изображений на основе нелинейных алгоритмов с использованием ранговой статистики. *Ярославский государственный университет имени П.Г. Демидова*, 2007.

ДОДАТКИ

ДОДАТОК А

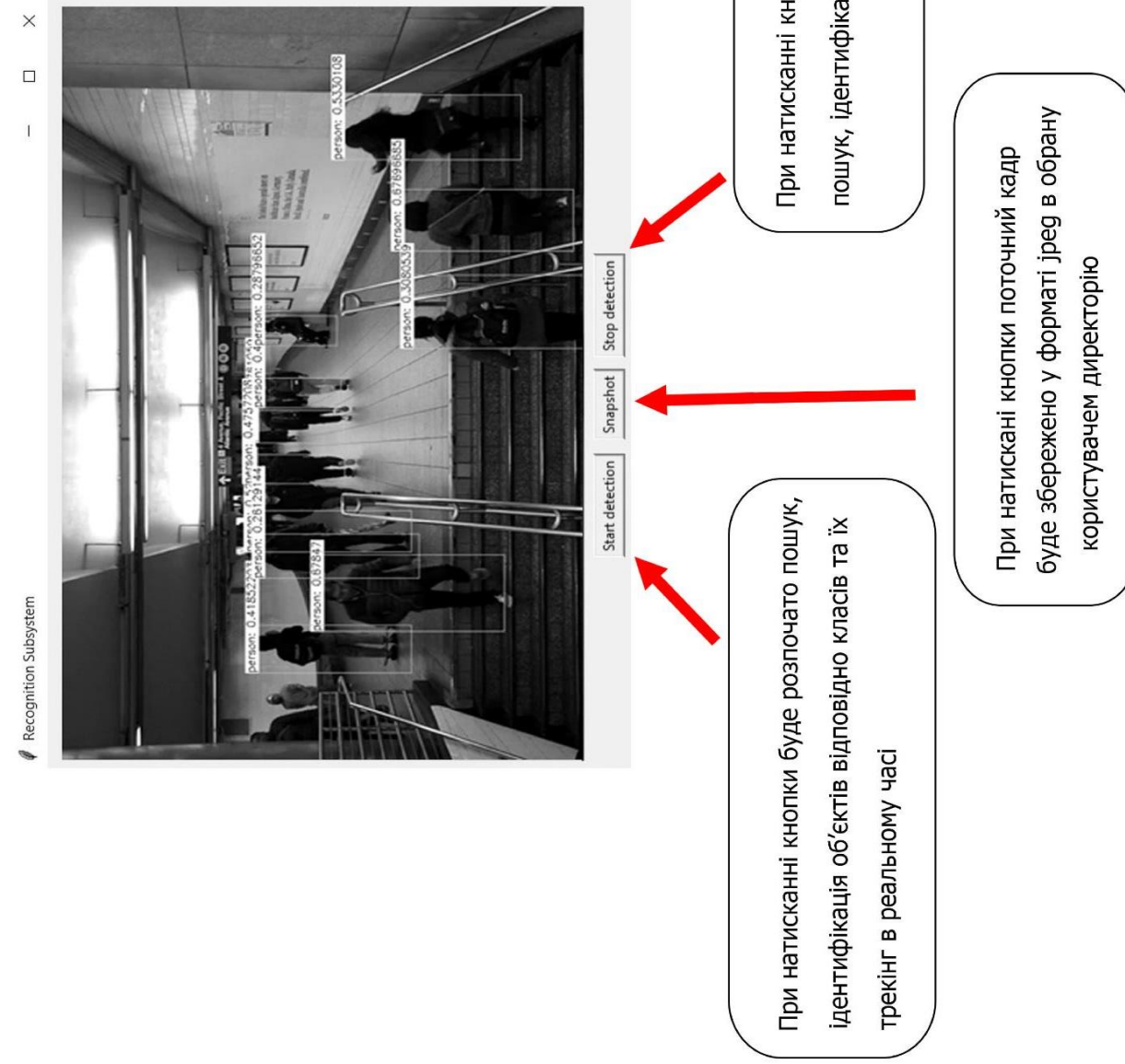
Плакати

ДОДАТОК Б

Перевірка на співпадіння

Інтерфейс програмного додатку у вигляді інструментарію для проведення досліджень

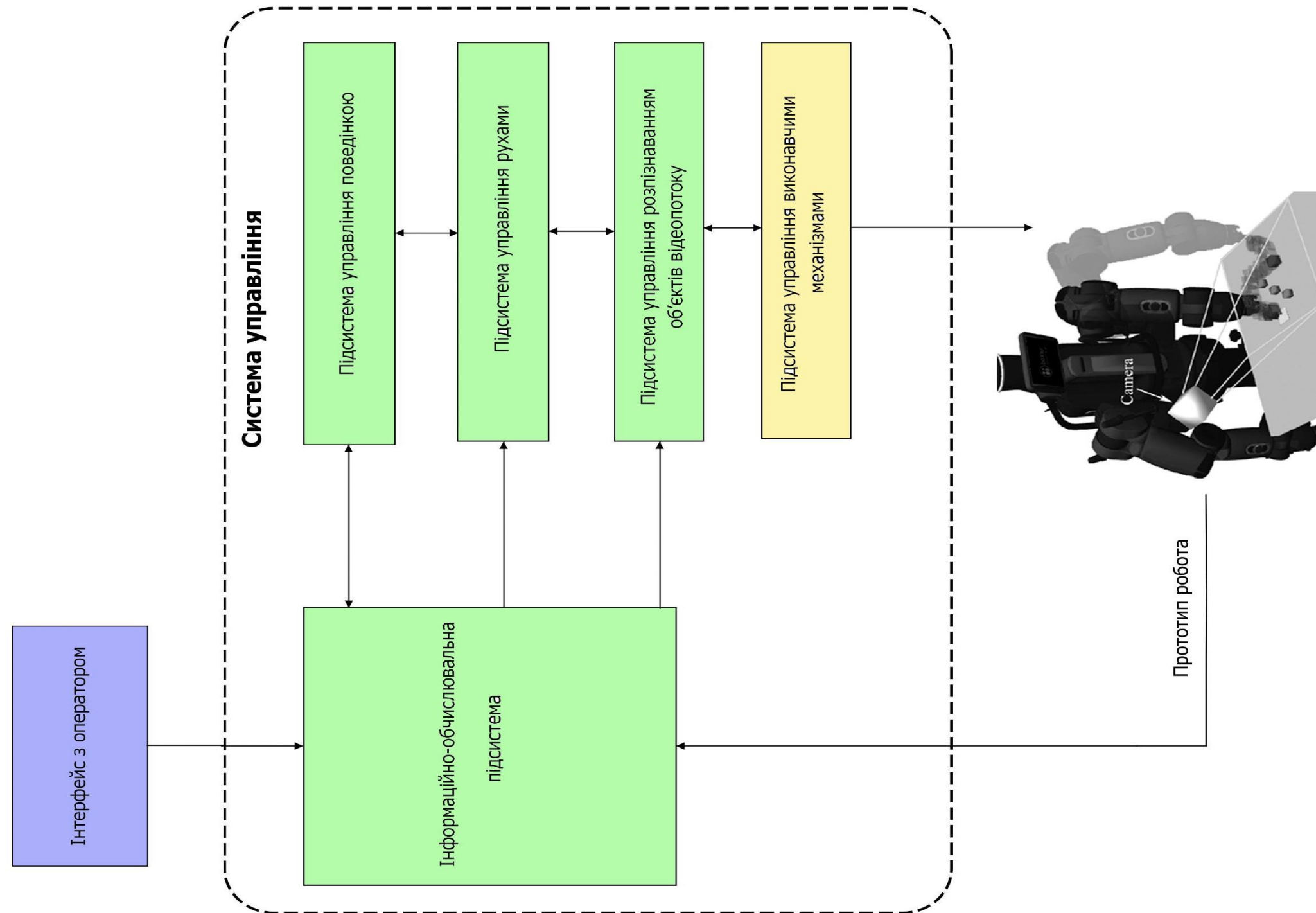
Показано приклад головного вікна розробленої підсистеми. Заголовок вікна містить назву програмного забезпечення "Recognition Subsystem". Внизу головного вікна розташовані кнопки "Start detection", "Snapshot" та "Stop detection".



Демонстраційний плакат №1
до магістерської дисертації на тему
«Підсистема розпізнавання об'єктів відеопотоку інтелектуального
робота»

Виконала: студентка гр. ІК-81МП Южда Ю. В.
Керівник: ст. викладач Анікін В. К.

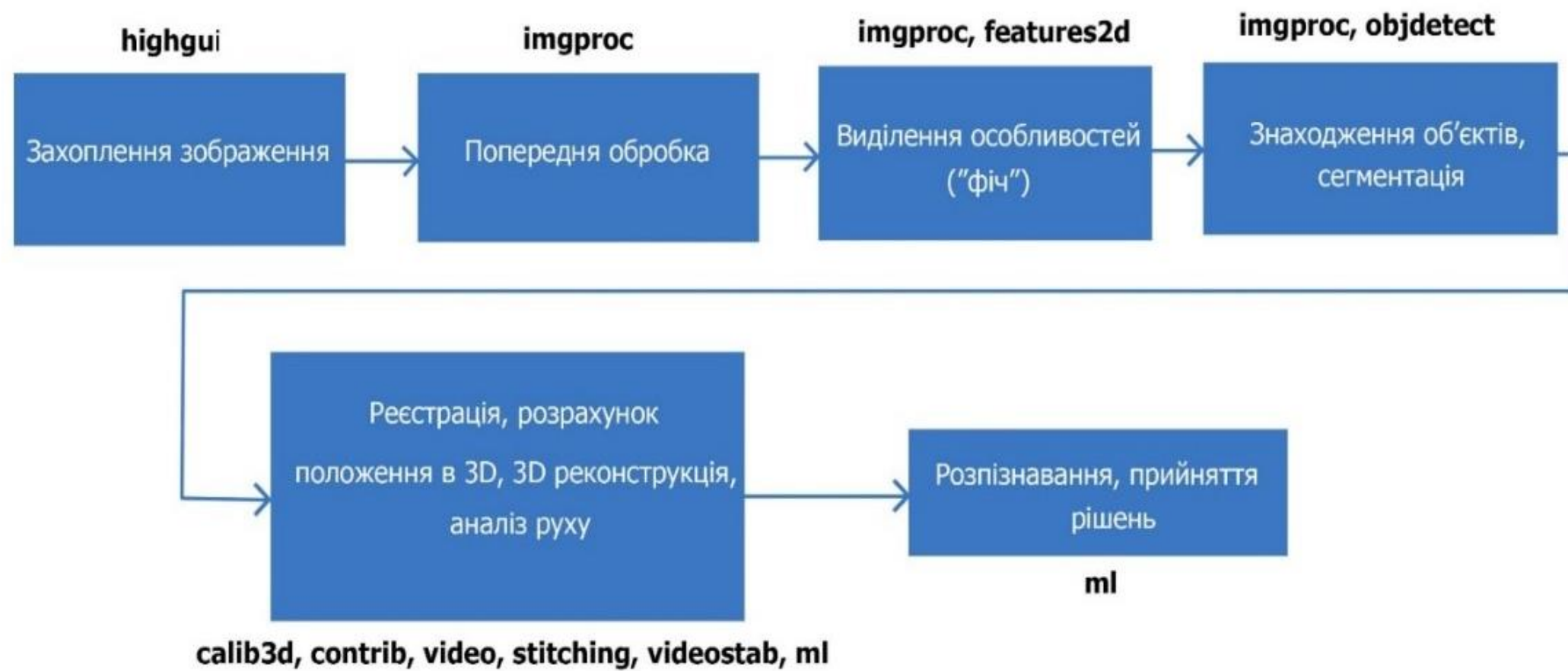
Система управління роботом



Демонстраційний плакат №2
до магістерської дисертації на тему
«Підсистема розпізнавання об'єктів відеопотоку інтелектуального
робота»

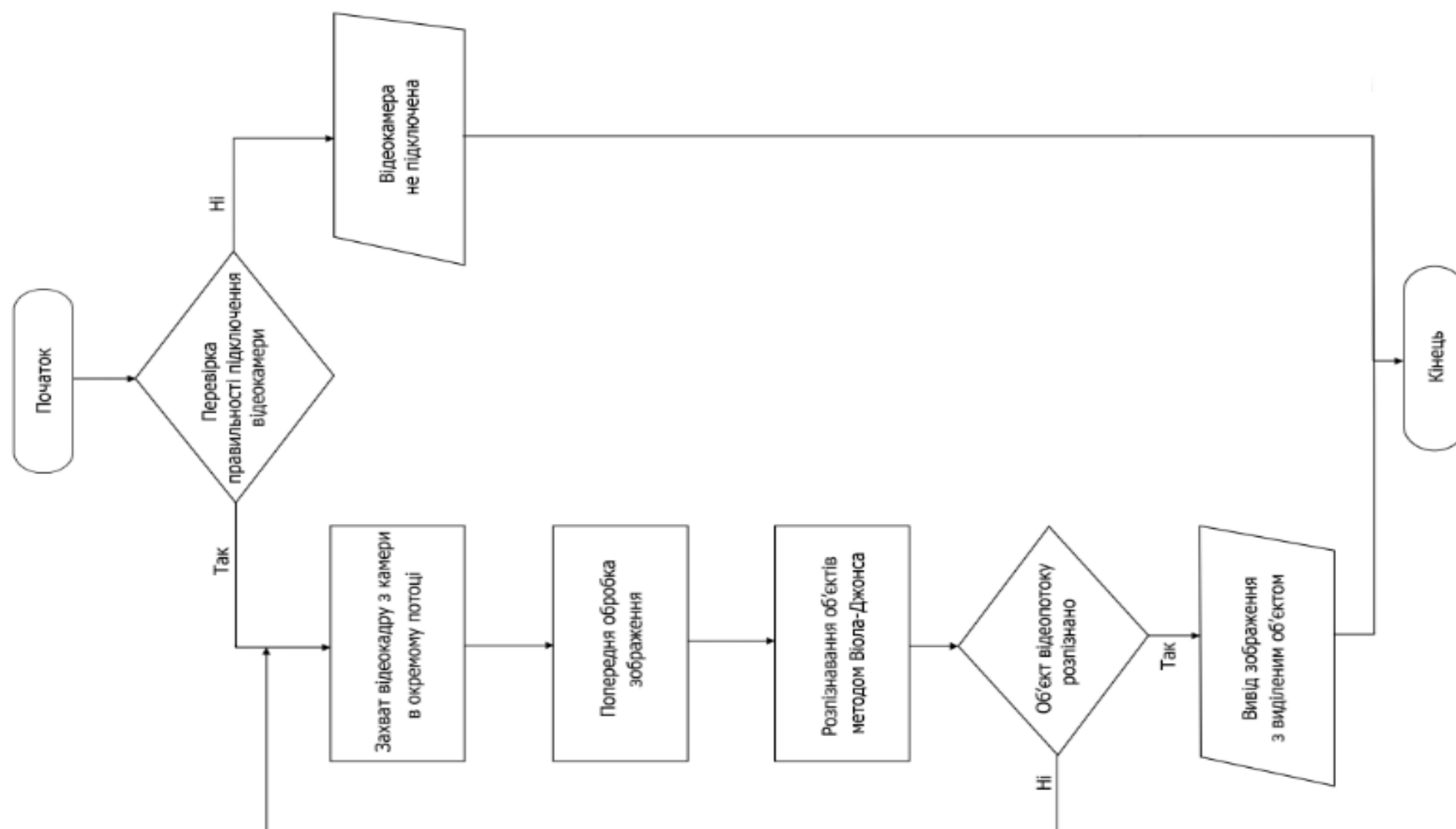
Виконала: студентка гр. ІК-81МП Южда Ю. В.
Керівник: ст. викладач Анікін В. К.

Модулі бібліотеки OpenCV



Демонстраційний плакат №3
до магістерської дисертації на тему
«Підсистема розпізнавання об'єктів відеопотоку інтелектуального
робота»

Виконала: студентка гр. ІК-81МП Южда Ю. В.
Керівник: ст. викладач Анікін В. К.



Демонстраційний плакат №4
до магістерської дисертації на тему
«Підсистема розпізнавання об'єктів відеопотоку інтелектуального
робота»

Виконала: студентка гр. ІК-81МП Южда Ю. В.
Керівник: ст. викладач Анікін В. К.

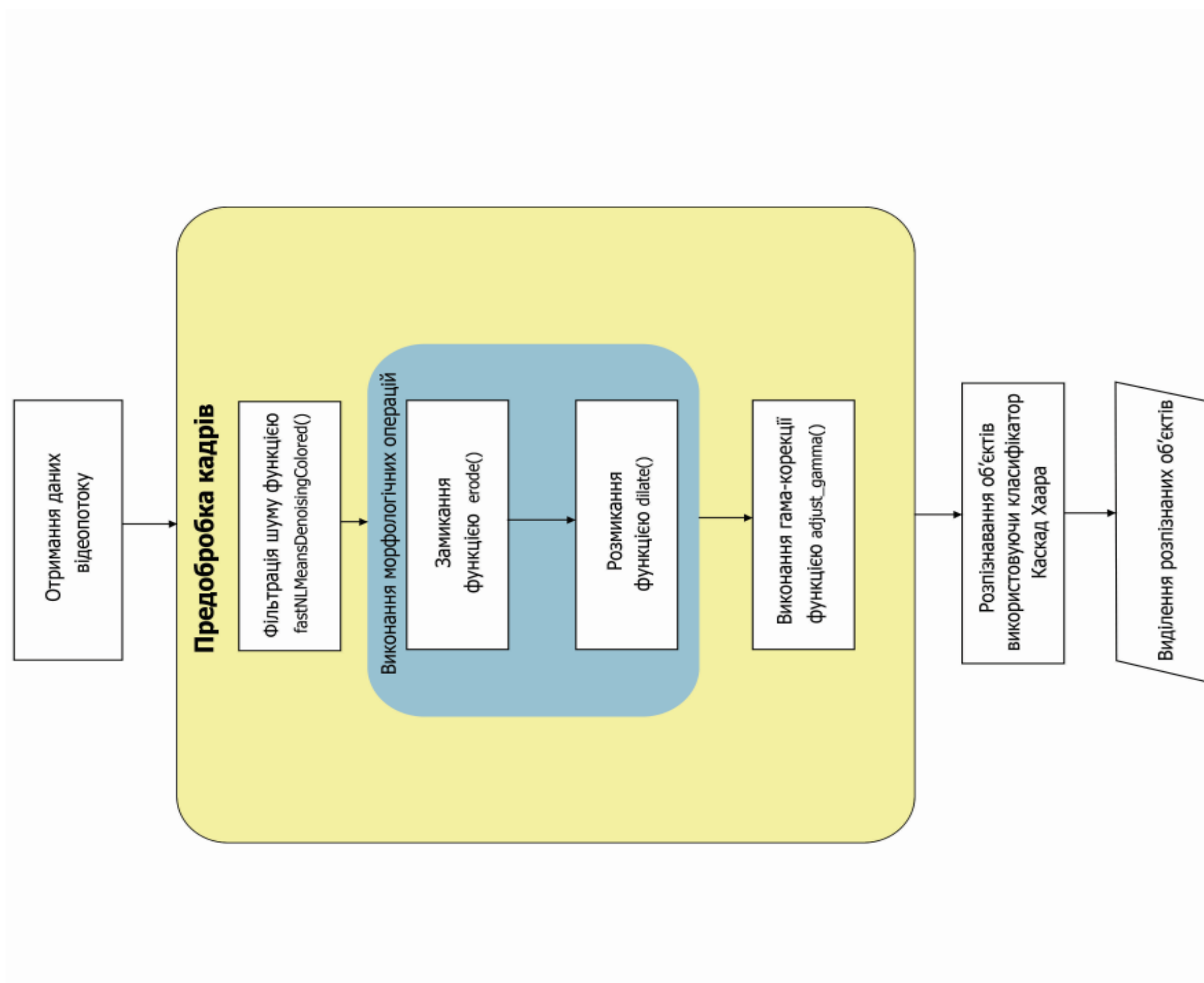
UML-діаграма програмного забезпечення



Демонстраційний плакат №5
до магістерської дисертації на тему
«Підсистема розпізнавання об'єктів відеопотоку інтелектуального
робота»

Виконала: студентка гр. ІК-81МП Южда Ю. В.
Керівник: ст. викладач Анікін В. К.

Схема попередньої обробки зображень



Демонстраційний плакат №6
до магістерської дисертації на тему
«Підсистема розпізнавання об'єктів відеопотоку інтелектуального
робота»

Виконала: студентка гр. ІК-81МП Южда Ю. В.
Керівник: ст. викладач Анікін В. К.